NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DETERMINATION OF DIESEL ENGINE CYLINDER GAS TORQUES FROM SPEED FLUCTUATIONS WITH A HIGH-FIDELITY CRANKSHAFT TORSIONAL MODEL

by

William J. Swanson

December 1998

Thesis Advisor: Knox T. Millsaps, Jr.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Master's/Mechanical Engineer Thesis

4. TITLE AND SUBTITLE
Determination Of Diesel Engine Cylinder Gas Torques From Speed Fluctuations With A High-Fidelity Crankshaft Torsional Model

5. FUNDING NUMBERS

6. AUTHOR(S)
Swanson, William J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

An experimental investigation was conducted to develop a method of predicting cylinder indicated torques in a reciprocating engine by measurement of crankshaft angular velocity fluctuations. Cylinder indicated pressures were measured for all three cylinders of a two-stroke Diesel engine with pressure transducers. Time-resolved angular position was measured at the crankshaft front and at the flywheel. A six degree-of-freedom torsional crankshaft model was developed. Two solution methods are described to solve the equations of motion: a time-marching ODE solver, and a Finite Element solution in the time domain. Using these methods with the measured cylinder torques, the angular positions are predicted and compared to measured angular positions for model calibration. An inverse solution method was developed to determine the cylinder indicated torques from the measured angular position at the crankshaft endpoints. The method is theoretically demonstrated to be useful for explicit solutions for two-stroke engines up to three cylinders, and four-stroke engines up to four cylinders. Experimental results show that the method is useful in predicting cylinder indicated torques from angular velocity measurements.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Diesel engine, Crankshaft angular velocity, Cylinder indicated pressure, Torsional model	124
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

DETERMINATION OF DIESEL ENGINE CYLINDER GAS TORQUES FROM SPEED FLUCTUATIONS WITH A HIGH-FIDELITY CRANKSHAFT TORSIONAL MODEL

William J. Swanson
Lieutenant, United States Navy
B.S., California Institute of Technology, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1998

ABSTRACT

An experimental investigation was conducted to develop a method of predicting cylinder indicated torques in a reciprocating engine by measurement of crankshaft angular velocity fluctuations. Cylinder indicated pressures were measured for all three cylinders of a two-stroke Diesel engine with pressure transducers. Time-resolved angular position was measured at the crankshaft front and at the flywheel. A six degree-of-freedom torsional crankshaft model was developed. Two solution methods are described to solve the equations of motion: a time-marching ODE solver, and a Finite Element solution in the time domain. Using these methods with the measured cylinder torques, the angular positions are predicted and compared to measured angular positions for model calibration. An inverse solution method was developed to determine the cylinder indicated torques from the measured angular position at the crankshaft endpoints. The method is theoretically demonstrated to be useful for explicit solutions for two-stroke engines up to three cylinders, and four-stroke engines up to four cylinders. Experimental results show that the method is useful in predicting cylinder indicated torques from angular velocity measurements.

TABLE OF CONTENTS

NOMENCLATURE

Symbol	Description	Units
A_p	Piston cross-sectional area	in^2
B	Cylinder bore	in
C	Torsional damping	lbf*in*sec/rad
[C]	Damping matrix	
D	Diameter	in
D_{cp}	Crankpin diameter	in
D_{jb}	Main journal bearing diameter	in
f	Friction factor	
F_{cr}	Connecting rod force	lbf
F_p	Net cylinder force from indicated pressure	lbf
F_r	Reaction force	
$\{F^e\}$	Element torque vector	
$\{F^{fe}\}$	Finite Element Method torque vector	
g	Gravitational acceleration	in/sec^2
h	Finite Element duration	sec
$\bar{h}$	Average bearing clearance	in
H_1, H_2	Shape functions	
I	Weighted average of residual	
[I]	Identity matrix	
J	Rotating mass polar moment of inertia	$lbf*in*sec^2$
J_{cp}	Crankpin rotating inertia	$lbf*in*sec^2$
J_{cr}	Connecting rod rotating inertia	$lbf*in*sec^2$
J_{jb}	Main journal bearing rotating inertia	$lbf*in*sec^2$
J_{rec}	Reciprocating mass polar moment of inertia	$lbf*in*sec^2$
[J]	Inertia matrix	
K	Torsional rigidity	lbf*in./rad
K_{gyr}	Radius of gyration	in
[K]	Torsional rigidity matrix	
$[K^e]$	Element matrix	
$[K^{fe}]$	Finite Element Method matrix	
L	Connecting rod length	in
L_e	Effective shaft length	in
L_{cp}	Crankpin length	in
L_{jb}	Main journal bearing length	in
N	Crankshaft rotational velocity	RPM
P_{cyl}	Cylinder indicated pressure (absolute)	lbf/in^2
P_{net}	Net pressure applied to cylinder	lbf/in^2
P_{ref}	Reference pressure (absolute)	lbf/in^2
Q	Flowrate	gpm
R	Crank throw	in
R_o	Counterweight outer radius	in
R_i	Counterweight inner radius	in
s	Piston linear position	in

$\dot{s}$	Piston linear velocity	in/sec
$\ddot{s}$	Piston linear acceleration	in/sec^2
t	time	sec
t_L	Finite Element initial time	sec
t_R	Finite Element final time	sec
T_{ct}	Counterweight thickness	in
T_{wb}	Crankweb thickness	in
T_{aux}	Auxiliary torque	in*lbf
T_{cyl}	Cylinder indicated torque	in*lbf
T_{load}	Load torque	in*lbf
T_{par}	Parasitic torque (piston friction)	in*lbf
T_{pmp}	Oil pump torque	in*lbf
T_{rec}	Reciprocating torque	in*lbf
V	Cylinder volume	in^3
w	Weighting function	
W	Weight	lbf
W_f	Bearing load	lbf
W_{rec}	Reciprocating weight	lbf
W_{wb}	Crankweb width	in
α	Counterweight angle	rad
ε	Phase deviation	rad
δ	Spring deflection	in
ϕ	Connecting rod angle at piston pin	rad
γ	Connecting rod angle at crankpin	rad
μ	Viscosity	reyns
θ	Crank Angle	rad
$\dot{\theta}$	Crankshaft rotational velocity	rad/sec
$\ddot{\theta}$	Crankshaft rotational acceleration	rad/sec^2
ρ	Specific weight	lbf/in^2
ω	Natural frequency	Hz
$\overline{\omega}$	Mean rotational velocity	rad/sec
BDC	Bottom dead center	
BHP	Brake horsepower	
CBM	Condition based maintenance	
ECA	Engine Cycle Analyzer	
FEM	Finite Element Method	
FFT	Fast Fourier Transform	
MDA	Modulation Domain Analyzer	
ODE	Ordinary Differential Equation	
RPM	Revolutions per minute	
RSM	Regularly scheduled maintenance	
TDC	Top dead center	
TTL	Transistor-to-Transistor Logic (signal type)	

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

A. MOTIVATION

Reciprocating engines require a maintenance system to ensure readiness throughout operational life. Currently, particularly in military uses, this is accomplished with a regularly scheduled maintenance (RSM) system, where parts are checked, and replaced if necessary, and components are serviced at intervals based on historical expected failure rates. Therefore, maintenance is normally performed well before it is actually necessary, because the true condition of the engine is unknown. Significant savings could be realized with the use of a condition based maintenance (CBM) system. Such a system requires a means to monitor engine health during operation, or with operational tests that do not require significant work on the engine.

Several classes of faults that occur in Diesel engines can be detected and localized by measurement of individual cylinder firing pressures. Examples include loss of compression ratio due to cylinder leaks, and improper combustion of fuel due to injector problems. Monitoring cylinder firing pressure is an excellent means of condition based maintenance. However, due to the harsh environment in the cylinders, the pressure transducers required are very expensive and short-lived. While direct measurement of cylinder pressures for performance monitoring is feasible and sometimes used in an operational engine, it is expensive. Of course, a possible solution to this problem would be the development of cheap, reliable pressure transducers for use in operational engines. Barring this, an alternative solution is the use of indirect methods for estimating cylinder

pressures, such as the measurement of crankshaft angular velocity fluctuations along with an appropriate scheme for inferring the pressure waveform.

B. STATE OF THE ART

The angular velocity of a reciprocating engine contains small fluctuations due to the variations of cylinder pressures. In general, the engine speeds up after a cylinder fires, then slows down as the next cylinder is compressing in preparation for combustion. The flywheel is intended to reduce the magnitude of these oscillations, but they are still present and represent a speed variation of several percent, which is a measurable amount. A number of researchers have investigated the possibility of predicting the cylinder pressure variation by measurement of these small speed oscillations.

Freestone and Jenkins [Ref 1] measured crankshaft velocity with a proximeter mounted at the flywheel ring gear teeth. They developed a lumped crankshaft model, using inertial torque to account for the reciprocating piston masses. This model was used to develop a calculation of the total gas torque in the engine cylinders as a function of crank angle. Noting abnormally low peaks in the pressure waveform and their corresponding crank angle localized faults in individual cylinders.

Mauer and Watts [Ref 2] measured angular velocity at both ends of the crankshaft by placing a proximeter at the flywheel ring gear teeth and at a corresponding gear mounted on the pulley. The phase difference between the two encoders corresponded to an instantaneous measurement of the total crankshaft twist, which was considered proportional to crankshaft torque. No mathematical model was used, so detection of faults was realized by comparing the measured signal to a signal recorded on a healthy engine. As expected, the twist signal had peaks that varied depending on which cylinder

was currently firing, because cylinders farther from the flywheel cause a greater twist for the same combustion pressure. Mauer continued this work [Ref 3], developing a lumped-parameter engine model to isolate cylinder specific torque. In this method, he computed the cylinder specific torque, defined as the integration of the total crankshaft torque from TDC of the cylinder in question to TDC of the next firing cylinder. In a healthy engine, cylinder torques will all be close to the mean, but a faulty cylinder will show a reduced specific torque compared to the other cylinders.

Citron, et al. [Ref 4] used a four degree-of-freedom model of the engine-drivetrain system that differentiated between the flywheel and the engine. Reciprocating masses were accounted for by an inertial torque component and the crankshaft speed was measured with a proximeter at the flywheel ring gear teeth. Total cylinder gas pressures were reconstructed by solving the equations of motion. Individual cylinder pressures were inferred by assuming that the majority of the net torque at a particular point was due to the cylinder undergoing the power stroke.

Connolly and Yagle [Ref 5] used a lumped engine model, assuming the total inertia of the crankshaft components as a single mass. An inertial torque accounted for reciprocating masses and the angular velocity was measured with a proximeter at the flywheel ring gear. A nonlinear differential equation relating combustion pressure to angular velocity, derived from the torque balance equation, was reformulated to a linear first-order differential equation relating pressure to the square of the angular velocity. Connolly revisited the issue [Ref 6] to reconstruct cyclic pressure variability from the crankshaft angular velocity.

Lim et al. [Ref 7] predicted cylinder pressures in a four-cylinder four-stroke spark ignition engine by making several assumptions about the cylinder pressure as a function of the crank angle. The intake and exhaust strokes were assumed reference values (intake manifold pressure and exhaust backpressure) and the compression stroke was estimated as a polytropic process for each cylinder. From knowledge of the crank angle and the firing order, power stroke pressures were estimated for each cylinder from the measured angular velocity and the known load torque. The method implied a lumped crankshaft model.

Iida et al. [Ref 8] measured angular velocity with a proximeter at the flywheel, and included a correction for tooth-to-tooth variation, determined by measurement of the flywheel at a constant rotational speed. A lumped engine model was used, in which an equation related the total engine inertia and rotational acceleration to the composite torque applied to the crankshaft. Integration of this equation over a cycle yielded a relation to determine Indicated Mean Effective Pressure (IMEP) from the total engine inertia and the square of the change in the angular velocity.

Taraza [Ref 9] developed a linear high-fidelity model of a multicylinder engine. Using this model, angular velocity and angular deflection were predicted for the front of the crankshaft and compared to measured values for an inline, four-stroke, four-cylinder Diesel engine, in order to verify the model parameters. He then determined harmonic orders for the crankshaft model, and conducted experimental measurements at these speeds. His results show good agreement between the measured and predicted harmonic order amplitudes. His conclusion was that the measured amplitudes of certain harmonic orders of angular motion could be used to determine engine mean indicated pressure.

Additionally, his work demonstrated the usefulness of a high-fidelity crankshaft torsional model.

Additional work on this subject [Refs 10-16] generally used a lumped crankshaft model with angular velocity measured at one point.

Previous work at NPS by Bell [Ref 17] and Hudson [Ref 18], on the same engine used in this study, demonstrated that there is information present in the crankshaft rotational speed of a reciprocating engine. Hudson developed a high-fidelity model of the engine crankshaft, then used measured pressures to predict speed fluctuations for comparison with actual speed fluctuations at the crankshaft nose. Due to noise from vibration in the optical encoder mounting, he was unable to show good agreement for speed fluctuations between measured results and the model.

To the best of the author's knowledge, no research has been reported using a high-fidelity torsional model to determine explicit cylinder-specific torques throughout a representative cycle. An engine crankshaft displays torsion that varies during a cycle, and along its length. This twist absorbs rotational energy that is later released when the twist relaxes. Previous models that consider the crankshaft as one rotating element neglect the effect this twist produces on the torques of individual cylinders. But when the intent is to localize engine faults, it is imperative that these differences between cylinders are considered.

C. OBJECTIVES

The primary objective of this study is to develop a method of determining individual cylinder gas torques from measured time-resolved angular positions at the two endpoints of the crankshaft, using a high-fidelity torsional model of the crankshaft.

Calibrated parameters will be determined for the torsional model from calculated values and experimental data.

Another objective is to develop numerical solution techniques for solving the equations of motion in both directions. Specifically, a method for direct integration of the differential equations of motion will be formulated that sets cyclic boundary conditions in the time domain. The reverse method, to determine cylinder gas torques from tine-resolved angular position data, must use data from only two measurement points instead of all degrees of freedom.

D. ORGANIZATION

Chapter II describes the development and calibration of the crankshaft torsional model. The equations of motion for the crankshaft are derived and presented.

Chapter III describes the experimental apparatus used to collect data for this study. Specifications for the test engine and diagrams for the instrumentation are presented. A summary of the various engine operating conditions used for the study is included.

In Chapter IV, an explanation of the numerical methods used to solve the equations of motion is described. This will include numerical methods for model calibration as well as cylinder gas torque prediction.

Chapter V shows the results of the engine test runs and data analysis that will support the thesis concept.

Chapter VI summarizes the study, presents conclusions, and lists recommendations for further research in this area.

II. CRANKSHAFT TORSIONAL MODEL

A. PHYSICAL SYSTEM

The engine used in this research is a Detroit Diesel (3-53 Series) three-cylinder two-stroke Diesel. The crankshaft (Figure 1) is supported by four main journal bearings, and includes counterweight lobes on four of the six crankwebs. The front of the engine is to the left in the diagram. A press fitted gear at the crankshaft nose drives the oil pump, and auxiliary loads are driven by the timing gear, located just forward of the flywheel (not shown).

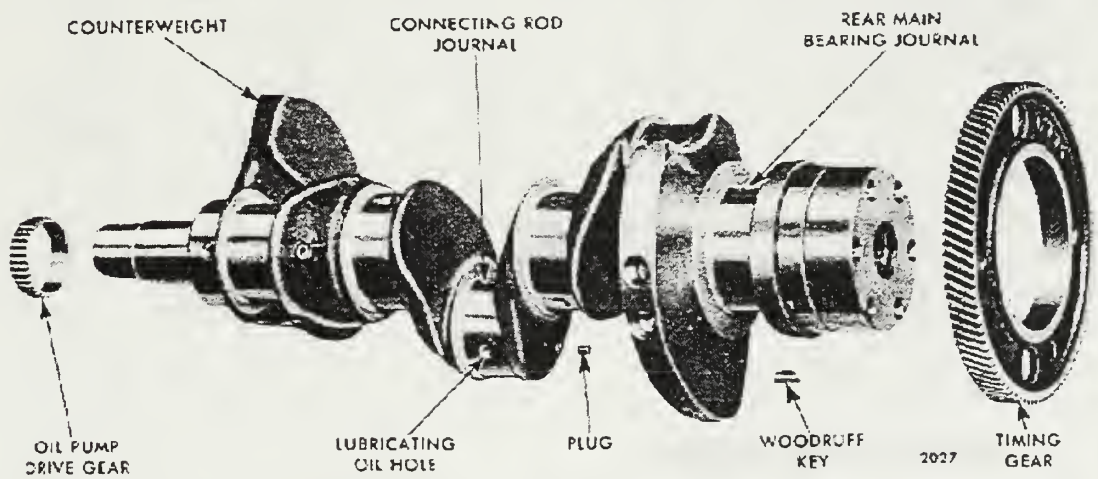


Figure 1. Crankshaft. From Ref [19].

An idealized mass-elastic torsional model is used to mathematically describe the angular motion of the crankshaft (Figure 2). This model, originally developed by Hudson [Ref 18], has been refined for the present study. Specifically, additional load torques were added to the model to account for the effect of the oil pump and the auxiliary loads, constant parasitic force was used to model the piston ring friction, and the effect of reciprocating torques was added. The model consists of six concentrated masses

connected by massless idealized shafting. The mass concentrations are centered at the optical encoder mounting, the three crankpins, the flywheel, and the dynamometer rotor. Torsional rigidity and damping are indicated by K and C, respectively. Gas torques and load torques are applied at the mass concentrations, as indicated in parenthesis.

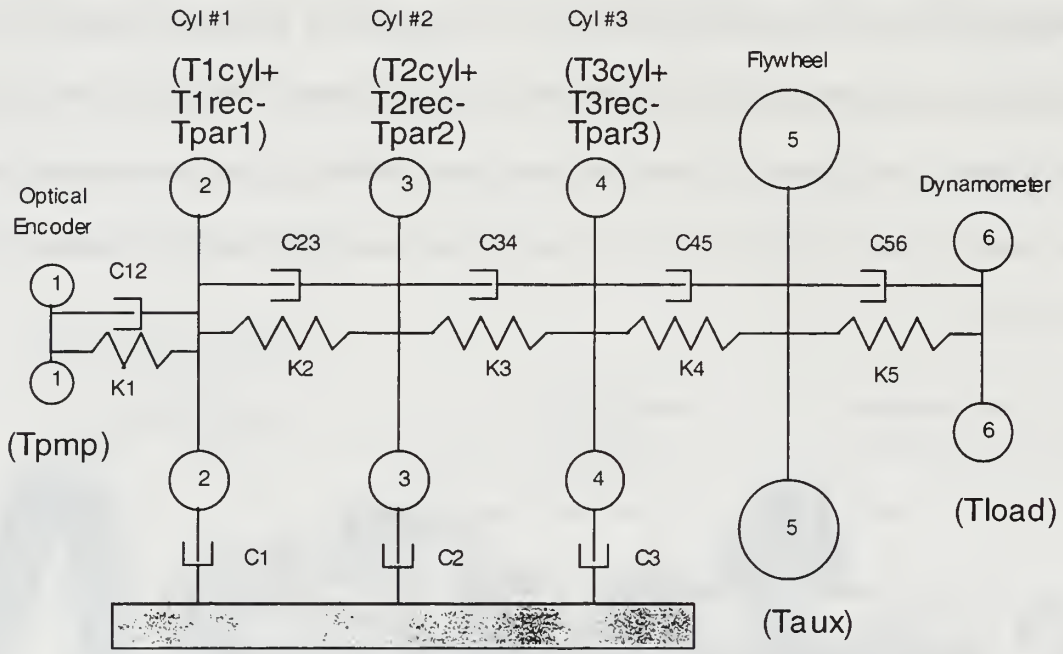


Figure 2. Crankshaft Torsional Model. After Ref [18].

B. EQUATIONS OF MOTION

Using the model from Figure 2, a set of six second order differential equations are developed to describe the rotational dynamics of the crankshaft. Since no part of the crankshaft is fixed, the model requires six separate angular position indications. These are the crank angles at each of the lumped mass points designated in the model, and they are designated θ_1 through θ_6.

$$J_1\ddot{\theta}_1 + C_{12}(\dot{\theta}_1 - \dot{\theta}_2) + K_1(\theta_1 - \theta_2) = -T_{pmp} \quad (1)$$

$$(J_2 + J_{2rec})\ddot{\theta}_2 + C_{12}(\dot{\theta}_2 - \dot{\theta}_1) + K_1(\theta_2 - \theta_1) + C_{23}(\dot{\theta}_2 - \dot{\theta}_3) + K_2(\theta_2 - \theta_3) + C_2\dot{\theta}_2 = T_{1cyl}(t) + T_{1rec}(t) - T_{par} \quad (2)$$

$$(J_3 + J_{3rec})\ddot{\theta}_3 + C_{23}(\dot{\theta}_3 - \dot{\theta}_2) + K_2(\theta_3 - \theta_2) + C_{34}(\dot{\theta}_3 - \dot{\theta}_4) + K_3(\theta_3 - \theta_4) + C_3\dot{\theta}_3 = T_{2cyl}(t) + T_{2rec}(t) - T_{par} \quad (3)$$

$$(J_4 + J_{4rec})\ddot{\theta}_4 + C_{34}(\dot{\theta}_4 - \dot{\theta}_3) + K_3(\theta_4 - \theta_3) + C_{45}(\dot{\theta}_4 - \dot{\theta}_5) + K_4(\theta_4 - \theta_5) + C_4\dot{\theta}_4 = T_{3cyl}(t) + T_{3rec}(t) - T_{par} \quad (4)$$

$$J_5\ddot{\theta}_5 + C_{45}(\dot{\theta}_5 - \dot{\theta}_4) + K_4(\theta_5 - \theta_4) + C_{56}(\dot{\theta}_5 - \dot{\theta}_6) + K_5(\theta_5 - \theta_6) = -T_{aux} \quad (5)$$

$$J_6\ddot{\theta}_6 + C_{56}(\dot{\theta}_6 - \dot{\theta}_5) + K_5(\theta_6 - \theta_5) = -T_{load} \quad (6)$$

For simplicity, the six equations can be combined into one matrix equation:

$$\begin{bmatrix} J_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_2 + J_{2rec} & 0 & 0 & 0 & 0 \\ 0 & 0 & J_3 + J_{3rec} & 0 & 0 & 0 \\ 0 & 0 & 0 & J_4 + J_{4rec} & 0 & 0 \\ 0 & 0 & 0 & 0 & J_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_6 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \end{Bmatrix} +$$

$$\begin{bmatrix} C_{12} & -C_{12} & 0 & 0 & 0 & 0 \\ -C_{12} & C_{12}+C_{23}+C_2 & -C_{23} & 0 & 0 & 0 \\ 0 & -C_{23} & C_{23}+C_{34}+C_3 & -C_{34} & 0 & 0 \\ 0 & 0 & -C_{34} & C_{34}+C_{45}+C_4 & -C_{45} & 0 \\ 0 & 0 & 0 & -C_{45} & C_{45}+C_{56} & -C_{56} \\ 0 & 0 & 0 & 0 & -C_{56} & C_{56} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{Bmatrix} +$$

$$\begin{bmatrix} K_1 & -K_1 & 0 & 0 & 0 & 0 \\ -K_1 & K_1+K_2 & -K_2 & 0 & 0 & 0 \\ 0 & -K_2 & K_2+K_3 & -K_3 & 0 & 0 \\ 0 & 0 & -K_3 & K_3+K_4 & -K_4 & 0 \\ 0 & 0 & 0 & -K_4 & K_4+K_5 & -K_5 \\ 0 & 0 & 0 & 0 & -K_5 & K_5 \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{Bmatrix} = \begin{Bmatrix} -Tpmp \\ T_{1cyl}(t) + T_{1rec}(t) - T_{par} \\ T_{2cyl}(t) + T_{2rec}(t) - T_{par} \\ T_{3cyl}(t) + T_{3rec}(t) - T_{par} \\ -Taux \\ -T_{Load} \end{Bmatrix}$$

or

$$[J]\ddot{\theta} + [C]\dot{\theta} + [K]\theta = \{T\} \quad (7)$$

C. CALCULATION OF MODEL PARAMETERS

The moments of inertia and the torsional rigidities can be analytically calculated for the model, as described in detail in Appendices A and B. It is assumed that the values for damping will be very low. Final values for the model are listed in Table 1.

Table 1. Model Parameter Values

	(lbf*in*sec^2/rad)		(10^6 lbf*in/rad)		(lbf*in*sec/rad)
J_1	0.02443	K_1	3.22	C_{12}	0.01
J_2	0.2482	K_2	7.00	C_{23}	0.01
J_3	0.1462	K_3	7.00	C_{34}	0.01
J_4	0.2482	K_4	10.82	C_{45}	0.01
J_5	7.2220	K_5	1.304	C_{56}	0.01
J_6	0.2870			C_2	0.013
J_{rec}	0.04955(1-cos2θ)			C_3	0.013
$J_{rec,avg}$	0.02478			C_4	0.013

Values for the friction and auxiliary loads are more difficult to determine analytically. Additionally, they will vary depending on the load and speed of the engine. Appendix A contains a theoretical analysis of friction and load torques, and this analysis was used to formulate an estimate of the expected magnitudes of friction and load torques. The values actually used in the model are listed in Table 2.

Table 2. Model Friction and Load Values

RPM	Load (ft*lbf)	T_{load} (in*lbf)	T_{pmp} (in*lbf)	T_{aux} (in*lbf)	F_{par} (lbf) (per cylinder)
1000	80	960	9.5	160	98
1000	100	1200	9.5	168	100
1000	135	1620	9.5	160	95
1500	135	1620	9.5	300	155
1500	160	1920	9.5	300	165
2000	160	1920	9.5	460	210

Values for the nonlinear model parameters (T_{rec}, J_{rec}, and T_{cyl}) are determined as functions of θ or t. Their derivation is described in detail in Appendices A and B.

III. EXPERIMENTAL METHODS

A. ENGINE DESCRIPTION

The Engine used in this research was a Detroit Diesel 3-53 Series engine, with characteristics listed in Table 3. For this study the front of the engine is designated as the end where the pulley would be located; the rear is the flywheel end. The cylinders are numbered consecutively from front to rear, so that cylinder #1 is the farthest from the flywheel. Cylinders are naturally aspirated; a roots blower provides a positive crankcase pressure that is proportional to engine speed [Ref 18]. The engine is considered to be a typical example of a Diesel engine.

Table 3. Engine Characteristics. From Ref [19]

Model	5033-5001N
Type	In-line two-stroke compression ignition
Number of Cylinders	3
Number of Main Bearings	4
Firing Order	1-3-2; Clockwise Rotation
Exhaust Valves per Cylinder	4
Displacement per Cylinder	53 in^3
Compression Ratio	21.0:1
Bore	3.875 in
Stroke	4.50 in
Max Rotation Speed	2800 RPM
Peak Torque	198 ft*lbf @ 1500 RPM
Max Power Output	92 BHP

The engine has been slightly modified. The front-end pulley was removed for mounting of the optical encoder to the crankshaft, and the alternator was removed (Figure 3). The engine was mounted on a Superflow engine test stand, and was loaded by an SF-901 Water Brake Dynamometer (Figure 4).

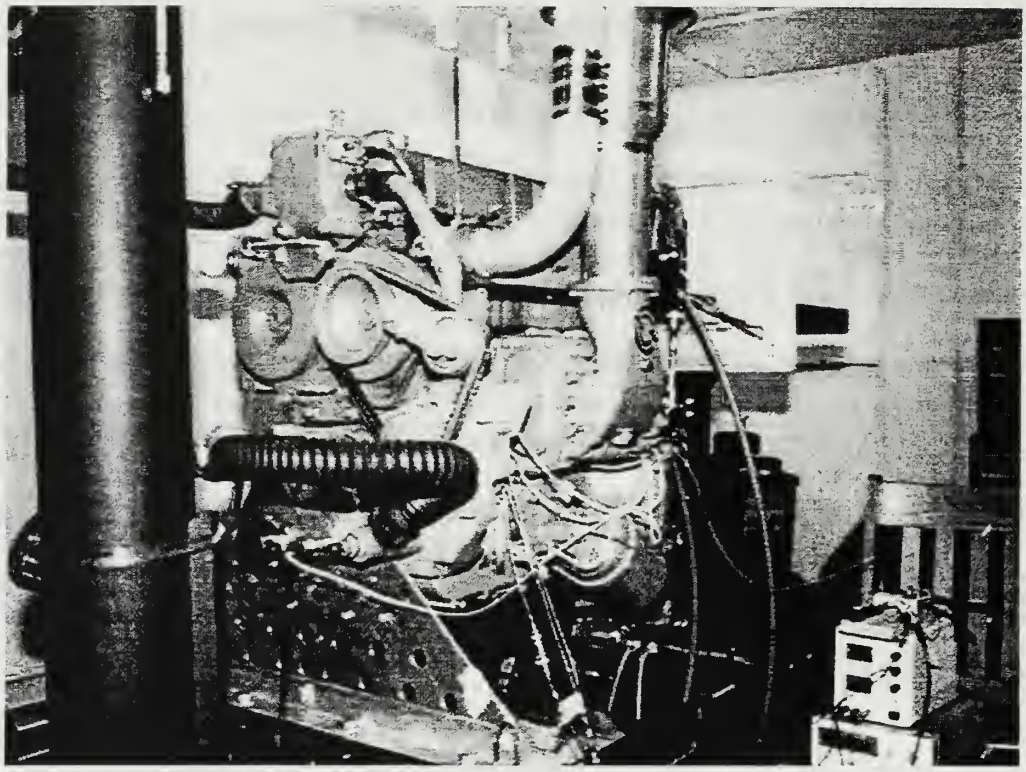


Figure 3. Engine Test Stand (Front View)

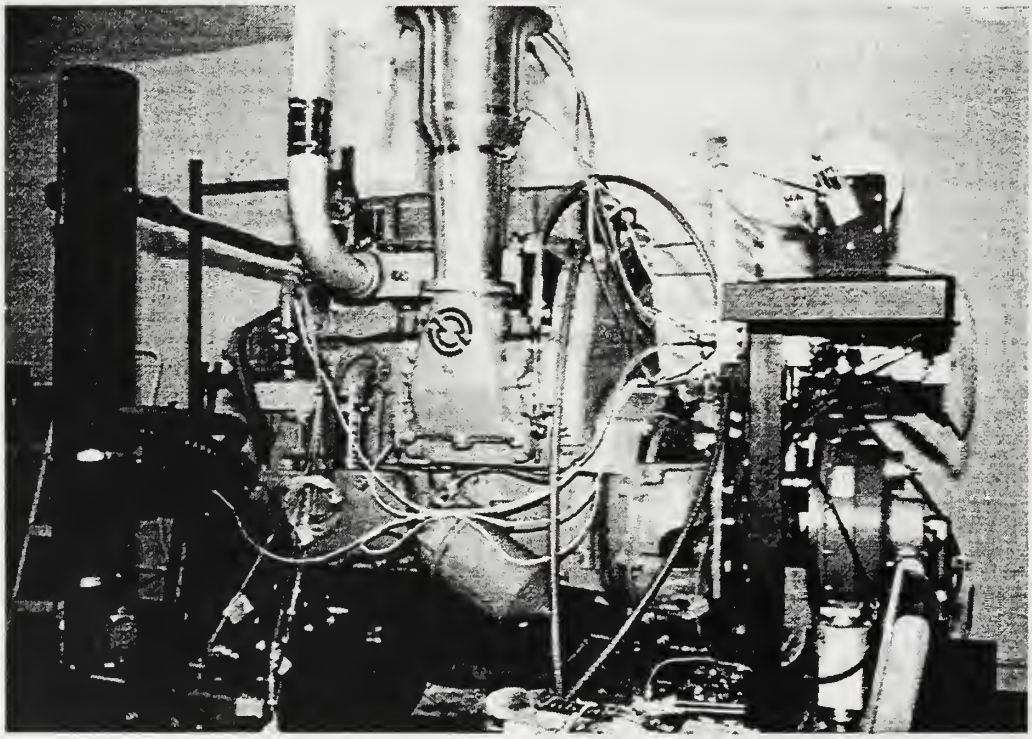


Figure 4. Engine Test Stand (Side View)

B. ENGINE CYCLE ANALYZER (ECA)

The Superflow Engine Cycle Analyzer (Figure 5) is a PC based data acquisition system. A sensor interface collects an engine load signal from the dynamometer, a crank angle and TDC signal from the optical encoder, and pressure signals from the piezoelectric pressure transducers mounted in the glow plug sockets for each cylinder. This information is passed to a data acquisition computer, which is used to store and display the pressure data. Raw pressure data are collected and phase-lock ensemble averaged over 11 cycles, then used in the numerical analysis programs. [Ref 20]

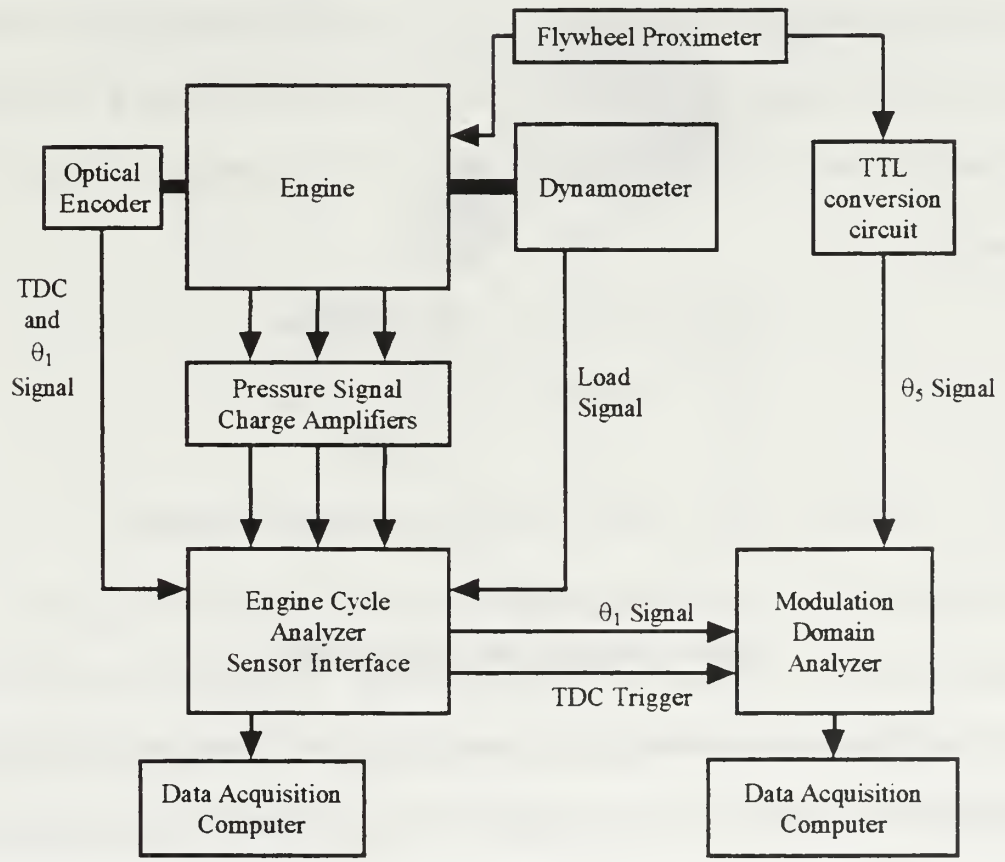


Figure 5. Instrumentation Schematic

C. OPTICAL ENCODER

A Heidenhain incremental Rotary Encoder [Ref 21] was used to collect time-resolved angular position at the crankshaft nose. The encoder consists of a flat optical disk, rigidly attached to the rotating shaft, with a specified number of evenly spaced windows etched near the perimeter (Figure 6). A signal is generated by photoelectric scanning of the disk as it rotates. The output signal is a TTL square wave, where highs correspond to a window passing in front of the detector. Measurement of the leading edge of the square wave corresponds to a time stamp for a specific angular position. The time differences inversely correspond to the average speed of the shaft as it rotates through the incremental angle. Encoders with 720 and 3,600 windows were available, but in either case 720 counts per revolution were collected, for an angular resolution of 0.5°. During a run data were collected for 11 cycles at the encoder for a total of 7,920 time stamps.

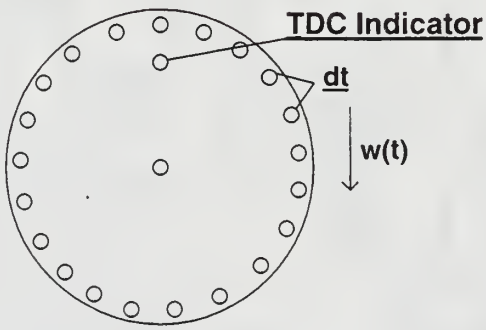


Figure 6. Optical Disk Representation

The Optical Encoder shaft was mounted to the end of the crankshaft with a flexible coupling (Figure 7). The coupling allows for radial and axial vibration of the crankshaft that would damage the encoder, because the endplay of the crankshaft exceeds the design specifications for the optical encoder without the protective coupling installed.

Hudson [Ref 18] collected his data with the encoder mounted directly to the crankshaft, resulting in extremely noisy data that did not compare favorably with predicted data. Additionally, excessive crankshaft radial vibrations damaged several encoders. The coupling transmits the angular position of the crankshaft nose to within an accuracy of 10" (4.85e-05 radians) [Ref 21]. An additional effect of the coupling is a high frequency torsional oscillation due to the natural frequency of the coupling/rotor combination. This is discussed in detail in Appendix C, and was not a significant problem since the signal of interest was at a much lower frequency.

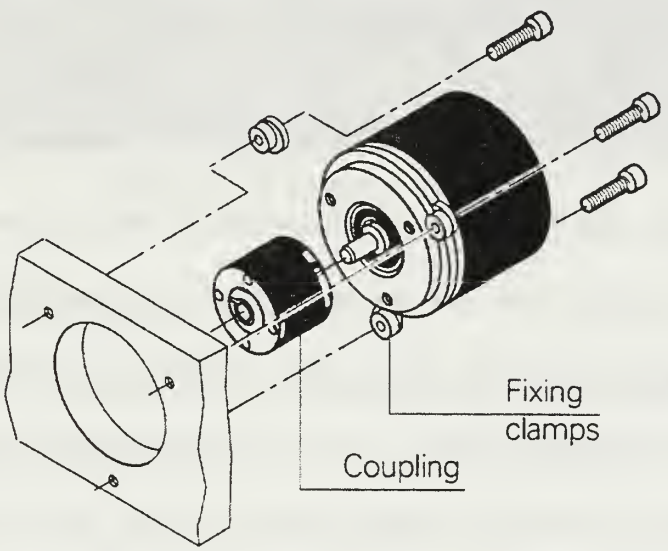


Figure 7. Optical Encoder and coupling. From Ref [21]

The body of the encoder is rigidly mounted to the engine block to minimize noise due to vibration (Figure 8). The mounting of the encoder is extremely important. Hudson used several different mounting schemes, before settling on the mount used again in this study, which works very well to ensure engine vibration does not affect the encoder measurements.

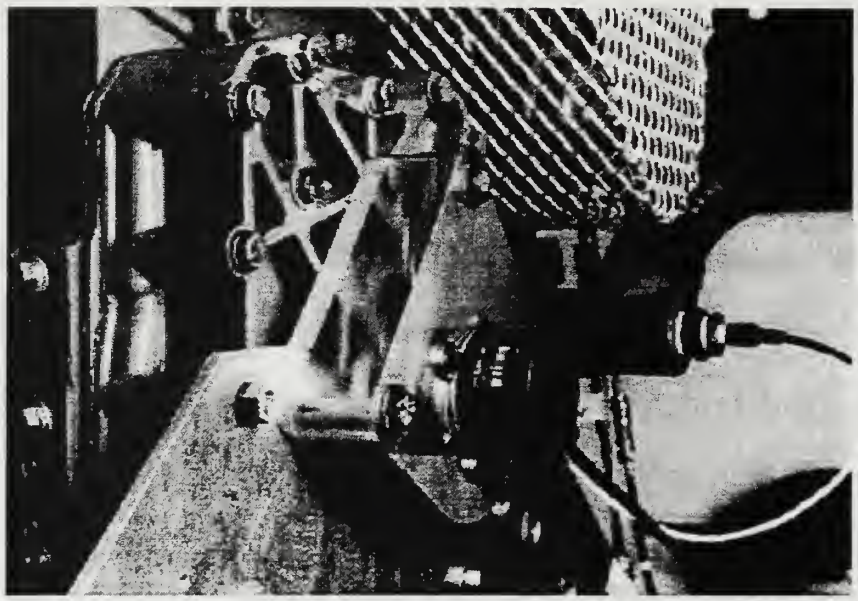


Figure 8. Optical Encoder Mounting. From Ref [18]

D. MAGNETIC INDUCTION PROXIMETER

A Bentley Nevada 3000 series 190 Proximitor system was used to detect passage of the flywheel ring gear teeth. The system consists of a ferromagnetic eddy current detector, which outputs a negative voltage that is a function of the distance between the probe end and a ferromagnetic surface. A TTL conversion circuit triggers a step change in voltage when the proximeter output exceeds a certain level, corresponding to a distance of about ¼ inch. The probe was mounted on a bracket fastened across the edge of the flywheel access panel, so that it saw the sides of the gear teeth as they passed (Figure 9). The output of the circuit was a square wave TTL signal; the leading edge of each wave corresponding to the passage of a gear tooth beneath the probe. The TTL output signal was sent directly to the MDA for data collection. There are 126 teeth on the flywheel ring gear, and during a run data were collected for 63 cycles, for a total of 7938 time stamps.

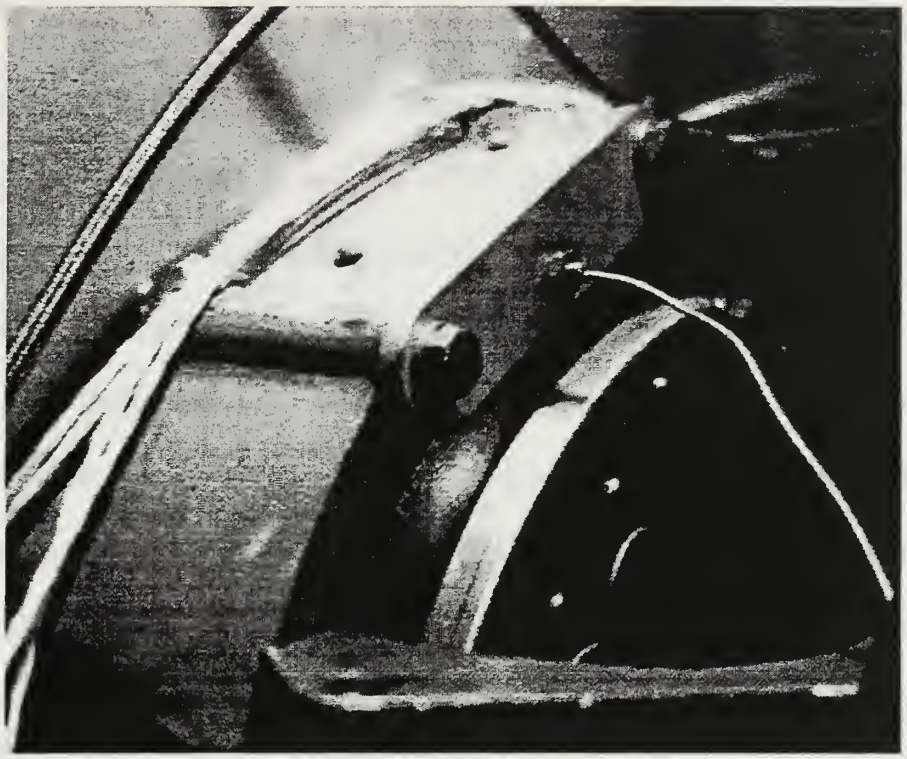


Figure 9. Proximeter mounting

E. MODULATION DOMAIN ANALYZER

A Hewlett Packard 53310A Modulation Domain Analyzer (MDA) was used to collect the time stamp data from the optical encoder and the proximeter. In either case, a TDC indicator (an output from the ECA) triggered the MDA. It received a TTL signal and recorded a time stamp at the leading edge of each wave. The MDA was able to collect up to 8,000 data points at a time. A single run collected 11 cycles from the optical encoder or 63 cycles from the flywheel proximeter, as described previously. A data acquisition computer controlled the MDA and received a transferred file containing the time stamps.

F. TEST MATRIX

A series of data runs were performed to test the validity and consistency of the model at varying engine speeds and loads. During a run data were collected for rotational speed at the flywheel and crankshaft nose, cylinder pressures for the three cylinders, dynamometer load, and atmospheric pressure. A data series was composed of data collected during a single run of the engine, at varying loads and speeds, normally taking about an hour on a single day. A prefix letter, such as "S," designated a particular series so that comparisons could be restricted to data taken during a single operation of the engine. This was intended to eliminate any variations in operation that might take place as the engine condition varied over time. Table 4 shows the elements of each data series, indicating what variations in load and speed make up each one. Table 5 lists specific information for each data run.

Table 4. Speeds and Loads for Data Series

ENGINE SPEED (RPM)	DYNAMOMETER LOAD (FT*LBF)				
	80	100	135	160	180
1000	S,U,V,W	S,T,U,V,W	S,T,U,V,W		
1500			S,T,U,V,W	S,T,U,V,W	
2000				T,U,V,W	V

Table 5. Data Run Information

DATA RUN	Date	Ambient Pressure (psia)	Comments
S	02 Sep 98	14.593	Heli – Cal coupling used; slippage of encoder shaft
T	11 Sep 98	14.662	Heli – Cal coupling used; TDC lag
U	08 Oct 98	14.819	Heidenhain K17 coupling used; TDC lag
V	14 Oct 98	14.730	Heidenhain K17 coupling used; TDC lag
W	21 Oct 98	14.706	Heidenhain K17 coupling used; no lag

IV. METHODS FOR CYLINDER PRESSURE AND TORQUE PREDICTION

The torsional model may be used to predict the motions of the crankshaft from measured applied forces, or it may be used to predict the forces from the measured motions. In order to do this, the differential equations of motion must be solved in both directions. First, two methods will be described for determining the angular positions θ_1 through θ_6, given the measured gas torques from the three cylinders. These solution methods, referred to as direct integration methods, will be used to test the validity of the torsional model and calibrate the parameters. Second, a method will be described for determining the individual cylinder gas torques T_{1cyl} through T_{3cyl}, given the time-resolved angular position at the two ends of the crankshaft, θ_1 and θ_5. This final solution method, called the inverse method, will be used for detection of cylinder faults from measurements of crankshaft rotational velocity.

A. PREDICTION OF PHASE DEVIATION FROM CYLINDER TORQUES

The equations of motion for the crankshaft model constitute a system of non-linear, second-order ordinary differential equations (ODEs). Calibration of the values for the torsional model will be conducted by first solving the differential equation for $\{\theta\}$, given the cylinder indicated torques. The predicted phase deviation and twist determined from $\{\theta\}$ can be compared to measured values to determine validity of the model.

1. Time-marching O.D.E. method

The first method is a direct integration of the ODEs in the time domain. This is accomplished numerically using a fourth- and fifth-order Runge-Kutte method. First, the six second-order ODEs must be converted to 12 first-order ODEs as follows:

$$\theta_7 = \dot{\theta}_1$$
$$\theta_8 = \dot{\theta}_2$$
$$\theta_9 = \dot{\theta}_3$$
$$\theta_{10} = \dot{\theta}_4$$
$$\theta_{11} = \dot{\theta}_5$$
$$\theta_{12} = \dot{\theta}_6$$
$$J_1\dot{\theta}_7 - C_{12}\,\theta_8 + C_{12}\theta_7 - K_1\theta_2 + K_1\theta_1 = -T_{pmp}$$
$$(J_2 + J_{2rec})\dot{\theta}_8 - C_{23}\theta_9 + (C_{23} + C_{12} + C_2)\theta_8 - C_{12}\theta_7 - K_2\theta_3 + (K_1 + K_2)\theta_2 - K_1\theta_1 = T_{1cyl}(t) + T_{1rec}(t) + T_{par}$$
$$(J_3 + J_{3rec})\dot{\theta}_9 - C_{34}\theta_{10} + (C_{34} + C_{23} + C_3)\theta_9 - C_{23}\theta_8 - K_3\theta_4 + (K_2 + K_3)\theta_3 - K_2\theta_2 = T_{2cyl}(t) + T_{2rec}(t) + T_{par}$$
$$(J_4 + J_{4rec})\dot{\theta}_{10} - C_{45}\theta_{11} + (C_{45} + C_{34} + C_4)\theta_{10} - C_{34}\theta_9 - K_4\theta_5 + (K_3 + K_4)\theta_4 - K_3\theta_3 = T_{3cyl}(t) + T_{3rec}(t) + T_{par}$$
$$J_5\dot{\theta}_{11} - C_{56}\theta_{12} + (C_{56} + C_{45})\theta_{11} - C_{45}\theta_{10} - K_5\theta_6 + (K_4 + K_5)\theta_5 - K_4\theta_4 = -T_{aux}$$
$$J_6\dot{\theta}_{12} + C_{56}\theta_{12} - C_{56}\theta_{11} + K_5\theta_6 - K_5\theta_5 = -T_{load}$$

(8)

These equations cannot be solved all in one step, however, because they are nonlinear due to the dependence of J_{rec} on θ, and because the T_{cyl}s and T_{rec}s are functions of time. Instead, a "time marching" method is used where the equations are solved over small steps and the final condition of each step becomes the initial condition for the subsequent step. The values of T_{cyl}, T_{rec}, and J_{rec} can then be approximated as a constant value over the step, or as a linear interpolation within the function describing the equation.

This method requires significant computing time. This is because a series of "shooting" iterations must be conducted to determine the initial conditions that yield cyclic conditions for the representative cycle (i.e., a periodic solution). Since the domain is an assumed representative cycle, it follows that the values of θ and $\dot{\theta}$ at the end of the cycle must match those at the beginning of the cycle. The initial angular velocities chosen will have a significant effect on whether or not the solution θ is "cyclic."

2. Finite Element method

Although the integration of the differential equations is an initial value problem, because it is assumed to describe a cyclic process the solution of the equation is known at future times. That is, at the end of one representative cycle, we expect all the angular positions to be increased by exactly 2π radians. An alternative method for solution avoids the shooting iterations by setting "boundary" conditions in time, instead of initial conditions. The problem is then treated as a boundary value problem in time, and a Finite Element Method (FEM) is developed to solve a second-order differential equation for θ as a function of t. This method is based on Kwon and Bang [Ref 22].

The weak formulation of the weighted residual method is used to approximate the solution to a second-order matrix differential equation. To accomplish this, the weighted average of the residual over the domain is set to zero:

$$I = \int_{t_0}^{t_f} w\left\{[J]\ddot{\theta} + [C]\dot{\theta} + [K]\theta - \{T\}\right\}dt = \{0\} \tag{9}$$

and then simplified to:

$$\int_{t_0}^{t_f} \left\{\left(w[C] - w[J]\right)\dot{\theta} + w[K]\theta\right\}dt = \int_{t_0}^{t_f} w\{T\}dt - \left\{w[J]\dot{\theta}\right\}_{t_0}^{t_f} \tag{10}$$

where w is the weighting function, and θ is a vector corresponding to the angular position at the six degrees-of-freedom.

A Galerkin Finite Element formulation is developed, using the sum of simple piecewise linear shape functions to approximate the more complex real function. The shape functions used are set up to be 1 at a node, linearly decreasing to 0 at the adjacent nodes. The value of the function at any point is approximated as a linear combination of

the values of the two shape functions for the adjacent nodes. Figure 10 shows these linear shape functions for a hypothetical element. The function T(t) is approximated as $T(t) = H_1(t)T(t_l) + H_2(t)T(t_r)$ between the nodes. This becomes the trial function for Galerkin's method, and the test functions are $w_1=H_1(t)$ and $w_1=H_2(t)$.

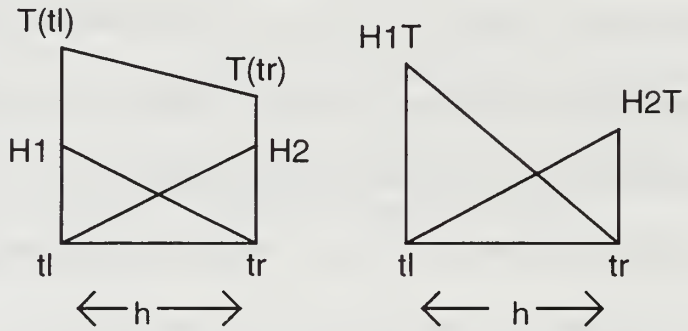


Figure 10. Linear Shape Functions. After Ref [22]

For a one-dimensional function, the shape functions are determined by:

$$H_1(t) = \frac{t_R - t}{h} \qquad H_2(t) = \frac{t - t_L}{h} \qquad h = t_R - t_L \tag{11}$$

But when used for a 6x6 matrix equation, the test functions must also be expanded into the following matrices:

$$[H_1] = H_1[I].....[H_2] = H_2[I].....[\dot{H}_1] = -\frac{1}{h}[I].....[\dot{H}_2] = \frac{1}{h}[I] \tag{12}$$

When substituted in Equation (10) and solved over the element from t_L to t_R, the result is:

$$[K^e]\begin{Bmatrix}\theta_L \\ \theta_R\end{Bmatrix} = \{F^e\} \tag{13}$$

where

$$[K^e] = \int_{t_L}^{t_R}\left(-\begin{Bmatrix}\dot{H}_1 \\ \dot{H}_2\end{Bmatrix}[J]\begin{bmatrix}\dot{H}_1 & \dot{H}_2\end{bmatrix} + \begin{Bmatrix}H_1 \\ H_2\end{Bmatrix}[C]\begin{bmatrix}\dot{H}_1 & \dot{H}_2\end{bmatrix} + \begin{Bmatrix}H_1 \\ H_2\end{Bmatrix}[K]\begin{bmatrix}H_1 & H_2\end{bmatrix}\right)dt \tag{14}$$

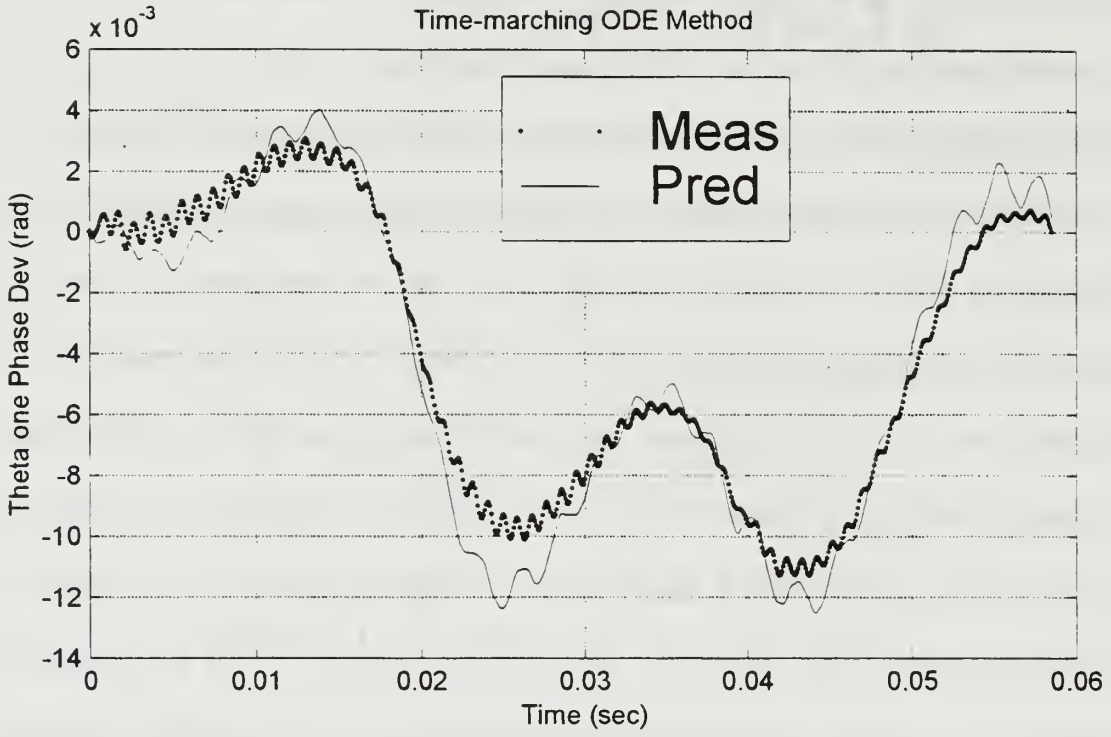
and

$$\{F^e\} = \int_{t_L}^{t_R} w\{T\}dt = \int_{t_L}^{t_R} \begin{Bmatrix} H_1 \\ H_2 \end{Bmatrix} \{T\}dt = \frac{h}{2}\begin{bmatrix} T_L \\ T_R \end{bmatrix} \tag{15}$$

Here, the element matrix $[K^e]$ is 12x12 and the element torque vector $\{F^e\}$ is 12x1. The values of the torque at the two nodes, $\{T_L\}$ and $\{T_R\}$, are 6x1 vectors. Equation (13) may be solved for the 6x1 vectors $\{\theta_L\}$ and $\{\theta_R\}$, which are the approximate solutions at the nodes. The solution used in this study has 720 elements and 721 nodes to solve for one representative cycle. The element matrices for each node are assembled into a 4326x4326 finite element matrix $[K^{fe}]$ and the element torque vectors are assembled into a 4326x1 finite element torque vector $\{F^{fe}\}$. Boundary conditions are established by defining the value of the 6x1 vector $\{\theta\}$ at the end nodes. This is accomplished by setting the first and last six lines of the finite element matrix $[K^{fe}]$ to identity, and setting the first six and last six values in the finite element torque vector $\{F^{fe}\}$ to the boundary values. The solution $\{\theta\}$ (a 4326x1 vector, the 720 6x1 nodal solutions $\{\theta_i\}$ stacked vertically) is then found from the matrix equation:

$$[K^{fe}]\{\theta\} = \{F^{fe}\} \tag{16}$$

This method shows a marked improvement in computing efficiency over the time-marching method. In addition to being about three times as fast for each program run, the Finite Element Method avoids the shooting iterations required for the time-marching method, which had to be repeated as many as five times for each solution. A comparison of the Phase Deviation found with both methods is shown in Figure 11. The Phase Deviation Plot will be described in detail in the next chapter.



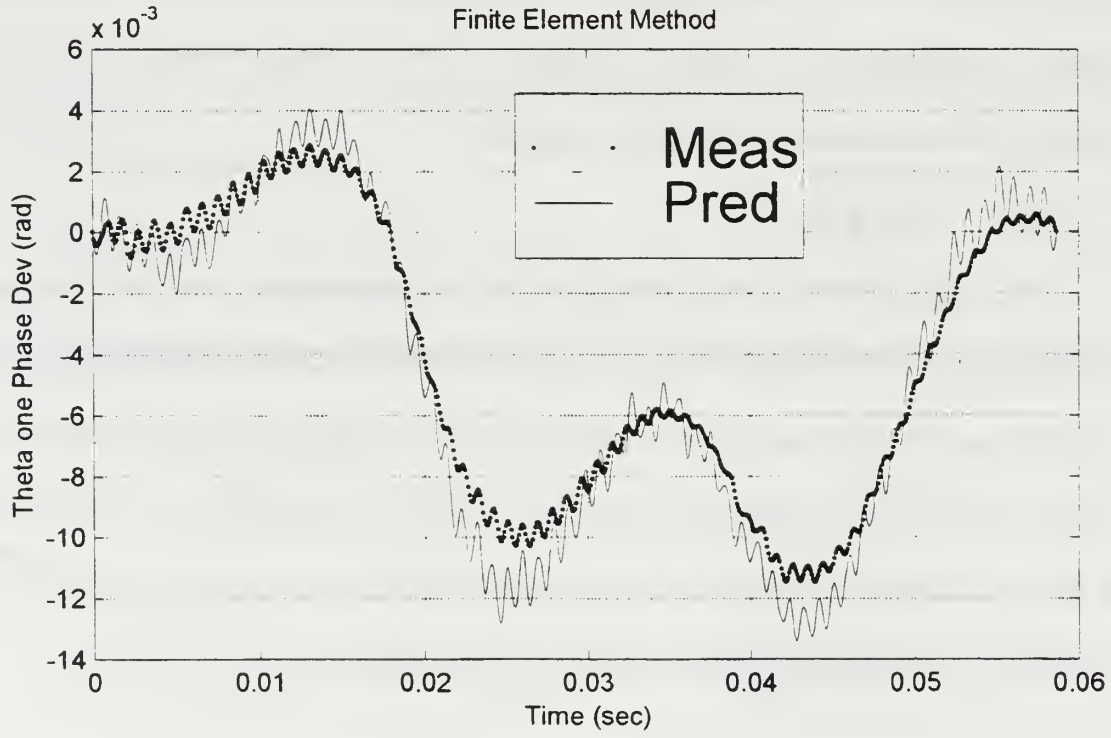


Figure 11. Comparison of Solution Methods

B. PREDICTION OF TORQUES FROM MEASURED SHAFT SPEEDS

1. Solution of Matrix Equation

a. *Two-Stroke Engines*

Given a properly calibrated model, the time dependent values of gas torque for the three cylinders (T_{1cyl}, T_{2cyl}, and T_{3cyl}) can be determined from the six second-order ODEs (Equations 1-6). Figure 12 outlines the solution method. By measurements at the flywheel and crankshaft nose, the values for θ_1 and θ_5 can be determined for a representative cycle. Numerical differentiation of this data yields the velocities and accelerations at these two points.

With properly calibrated parameters for the torsional model, Equation (6) can be solved for θ_6, then Equation (5) can be solved for θ_4, and Equation (1) can be solved for θ_2. Now there are four unknowns left (θ_3, T_{1cyl}, T_{2cyl}, and T_{3cyl}) and three equations (Equations 2, 3, and 4). However, because this is a two-stroke engine, each cylinder is at reference pressure for one third of each cycle, while the exhaust and intake ports are uncovered. Therefore, at any one time during a representative cycle, one of the three cylinder torques is known, leaving three unknowns and three equations. For example, for the first 120 degrees of the cycle, cylinder #2 ports are open, so the pressure in cylinder #2 is reference pressure (See Figure 13). For the first 120 degrees, T_{2cyl} can be calculated from this reference pressure. Then θ_3 can be calculated from Equation (3), and T_{1cyl} and T_{3cyl} can be determined from Equations (2) and (4). The other two-thirds of the representative cycle are solved in a similar manner.

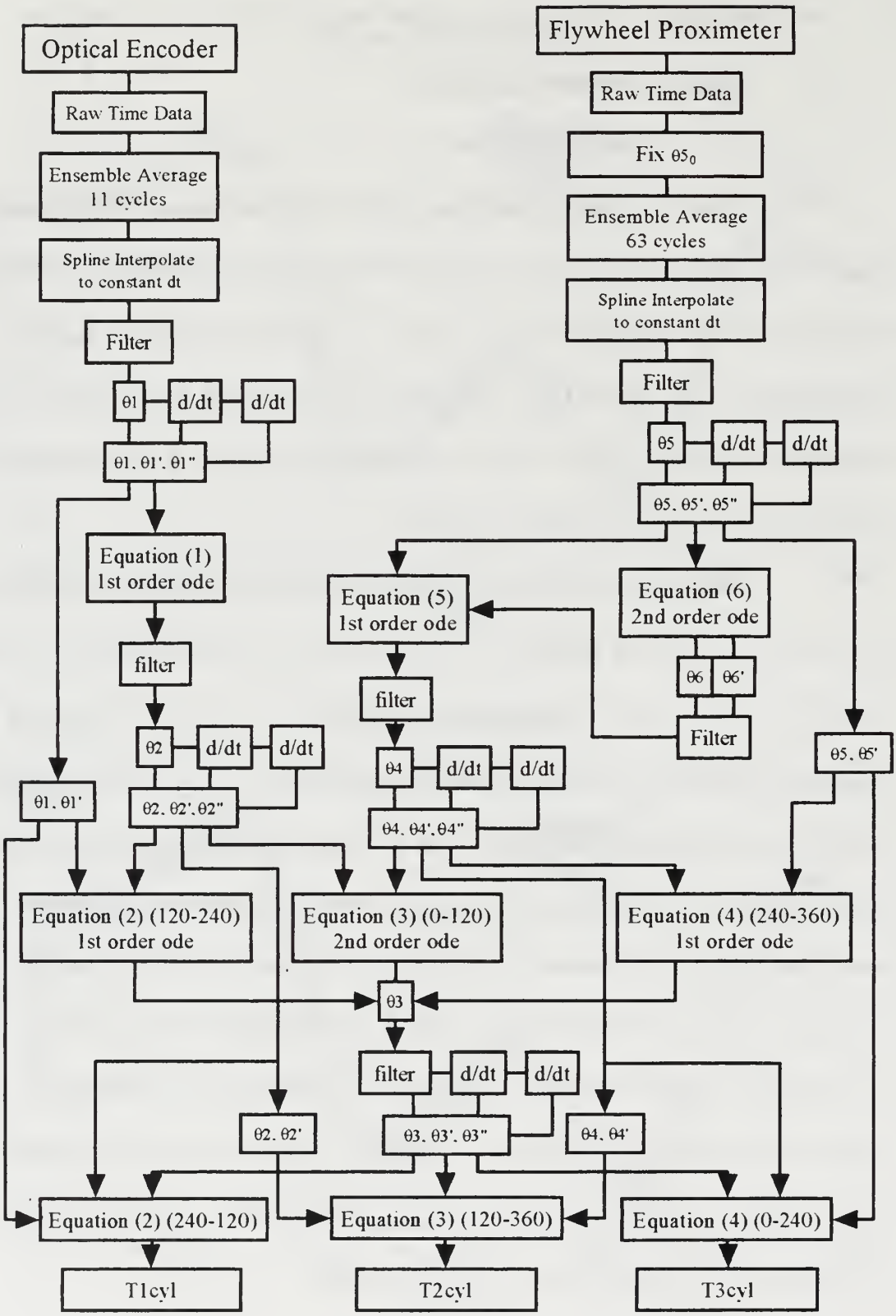


Figure 12. Torque Prediction Flowchart

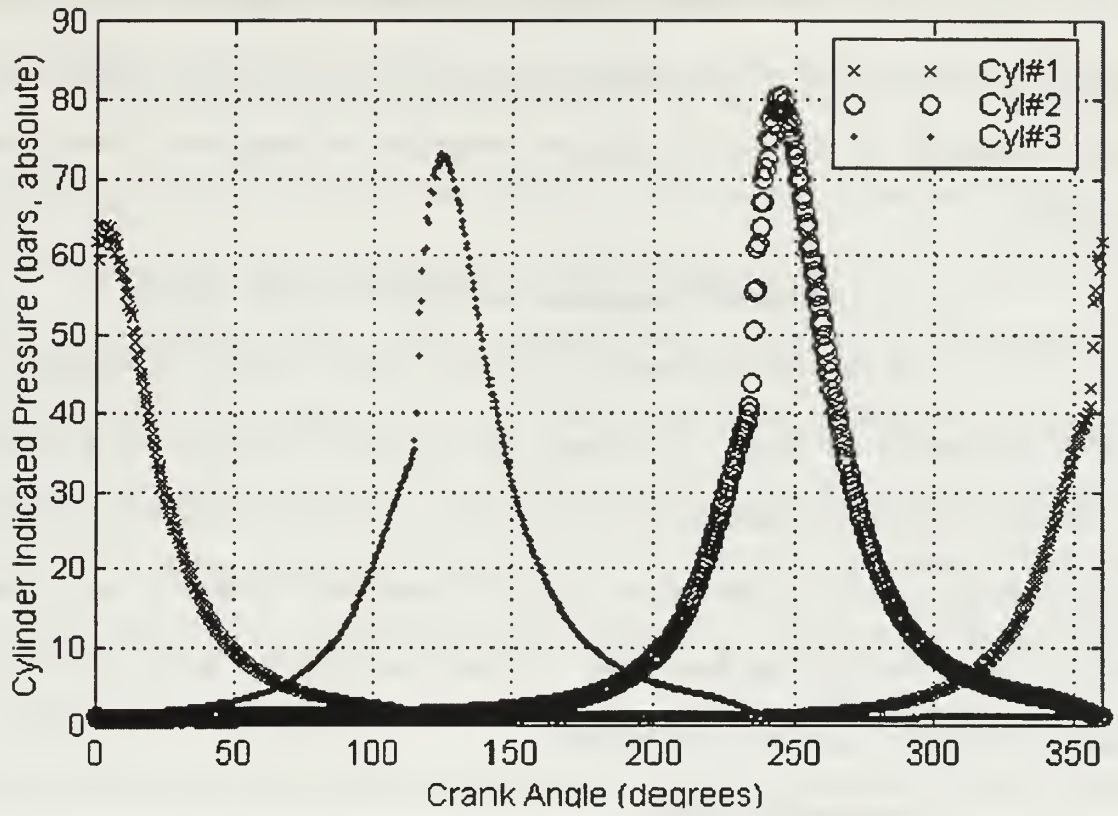


Figure 13. Cylinder Gas Pressures (1000 RPM, 100 Ft*lbf)

Although the ports are not actually uncovered for the first and last 10 degrees, measurements show the assumption of reference pressure is reasonable for the entire 120 degrees. The assumptions used here limit the feasibility of this solution method to two-stroke engines with three or fewer cylinders.

b. ***Four-Stroke Engines***

For a four stroke engine, two full rotations of the crankshaft must be considered to cover the power, exhaust, intake, and compression strokes. Over the representative two rotation (one cycle) period, a cylinder's pressure can be assumed to be equal to intake manifold pressure during the intake stroke, and exhaust back pressure during exhaust stroke. Therefore, for a particular cylinder, torque is known for half of the

cycle. The solution method above is feasible for four-stroke engines with four or fewer cylinders. A four cylinder engine would require a seven degree of freedom model, which could be explicitly solved as above, using the assumptions for intake and exhaust stroke pressure.

c. *Additional Assumptions for Multiple Cylinder Engines*

Engines with more than four cylinders can be analyzed by this method if further assumptions are made. For instance, the cylinder compression stroke can be estimated as a polytropic compression of an ideal gas. For a large engine with cylinders that fire simultaneously, the two cylinders could be lumped and considered as one inertial mass. Also, a measurement device may be placed internal to the engine to measure angular velocity at a third degree-of-freedom.

2. Interpolation of Data

The rotational speed data that are collected at the optical encoder and the flywheel consists of information that is uniformly spaced in the angular position domain, not in the time domain. This arises because of the nature of the data collection (See Section III.C and III.D). The raw data for $\theta_1(t)$ and $\theta_5(t)$ are converted to values which are evenly spaced in the time domain for further numerical analysis (specifically, this is useful for filtering; see section IV.B.3). This requires interpolation of the raw data. Interpolation is accomplished numerically using a cubic spline method. This means that the curve is assumed to be a 3^{rd} order polynomial with continuous slope at each of the data points. Interpolated values of $\theta(t)$ are determined for a selected evenly-spaced time basis.

3. Signal Filtering by Fast Fourier Transform

The measured data for position, θ(t), and angular velocity, $\frac{d\theta(t)}{dt}$, contain high frequency components (where "high frequency" refers to frequencies much higher than about three times the rotational speed). These frequency components are due to high frequency torsional vibration of the crankshaft at the various natural frequencies, and random noise from unknown sources. For solution of the equations necessary to predict torques, the position data must be differentiated once to determine angular velocity and twice to determine angular acceleration (see section IV.B.1). If raw data were used in the analysis, the high frequency components would be greatly amplified by subsequent differentiation. However, the torques of primary interest in this problem oscillate at about three times the rotational velocity of the engine. Therefore, the much higher frequency components are filtered out before differentiation in order to increase the signal-to-noise ratio in solving the equations for torque.

A phase deviation signal is derived from the raw data by comparing it to the mean rotational speed. This phase deviation signal is then filtered numerically using a Fast Fourier Transform (FFT), removing the high frequency data, and then performing an inverse FFT to achieve the desired filtered phase deviation signal. The cut-off frequency typically used was between six and nine times the rotational speed of the engine.

4. Numerical Differentiation

Once the angular position data has been interpolated and filtered, it must be differentiated once to obtain angular velocity and twice to obtain angular acceleration as functions of time. A central difference technique is used, where the numerical derivative at a point is the sum of the two adjacent differences divided by twice the time difference.

The endpoints are exceptions; they are determined by the single adjacent difference divided by the time difference (essentially a forward difference for the first point and a backward difference for the last point). No further filtering is required after differentiation.

C. TEST OF SOLUTION METHODS

The methods previously discussed are used to solve the equations of motion in both directions. Using a set of measured cylinder indicated pressures, the accuracy of the numerical methods can be tested. The time-marching ODE method was used to solve for the crankshaft time-resolved angular positions from the measured cylinder indicated torques. The predicted θ_1 and θ_5 values were then used in the inverse solution method to predict the cylinder indicated torques. Comparison of the resulting predicted torques to the original measured torques can be used to quantify the accuracy of the numerical solution methods. The test results for individual cylinder gas torques are shown in Figure 14, and the results for total gas torque is shown in Figure 15. These results show that the numerical methods tend to introduce a 2% peak-to-peak error and a 4 degree lag in the predicted total gas torque.

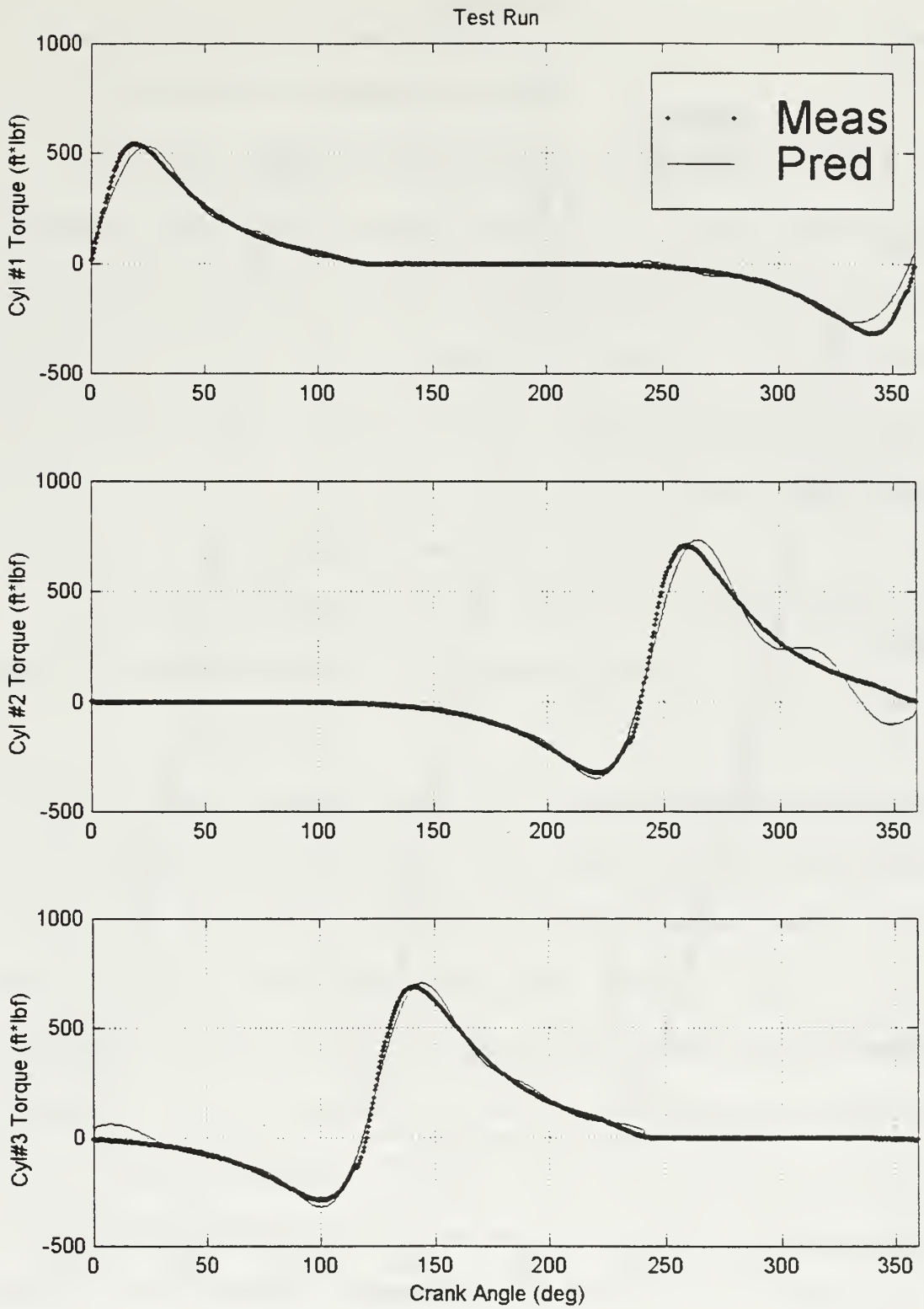


Figure 14. Test of Numerical Methods for Individual Torques

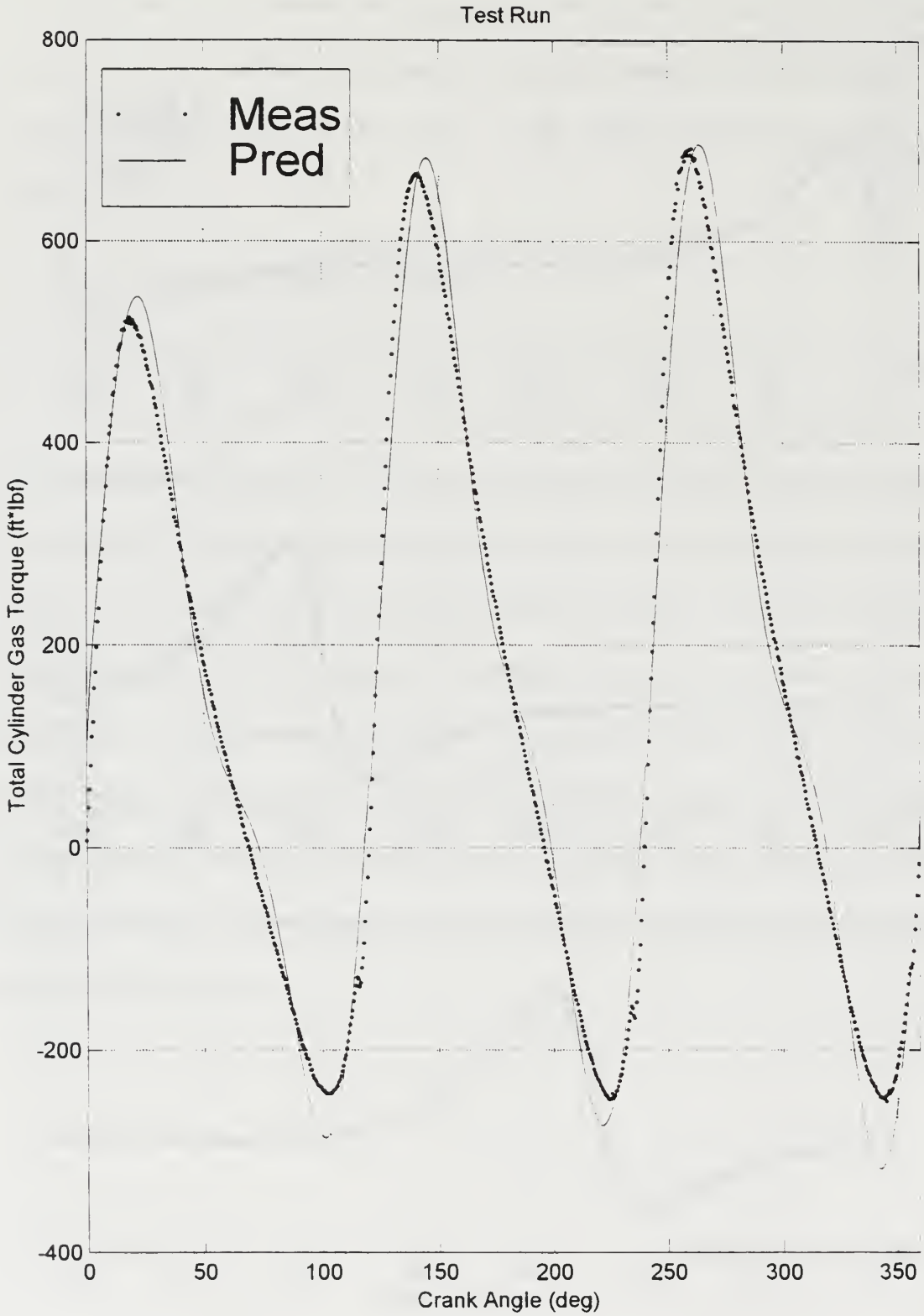


Figure 15. Test of Numerical Methods for Total Torque

V. RESULTS

A. CALIBRATION OF INERTIAL MODEL

Although the stiffness and inertial parameters of the equations of motion can be analytically estimated, some fine-tuning of the values is required to ensure the model matches the actual crankshaft. Given the measured torques in the three cylinders, and the methods of the previous chapters, the equations of motion may be solved for θ_1 through θ_6. Experimental data is collected for the two measurement points, θ_1 and θ_5. Two useful comparisons between the measured and predicted time-resolved angular positions are the Phase Deviation Plot and the Crankshaft Twist Plot.

The Phase Deviation Plot shows the oscillation of the angular position about a theoretical mean rotating position. It is calculated as follows:

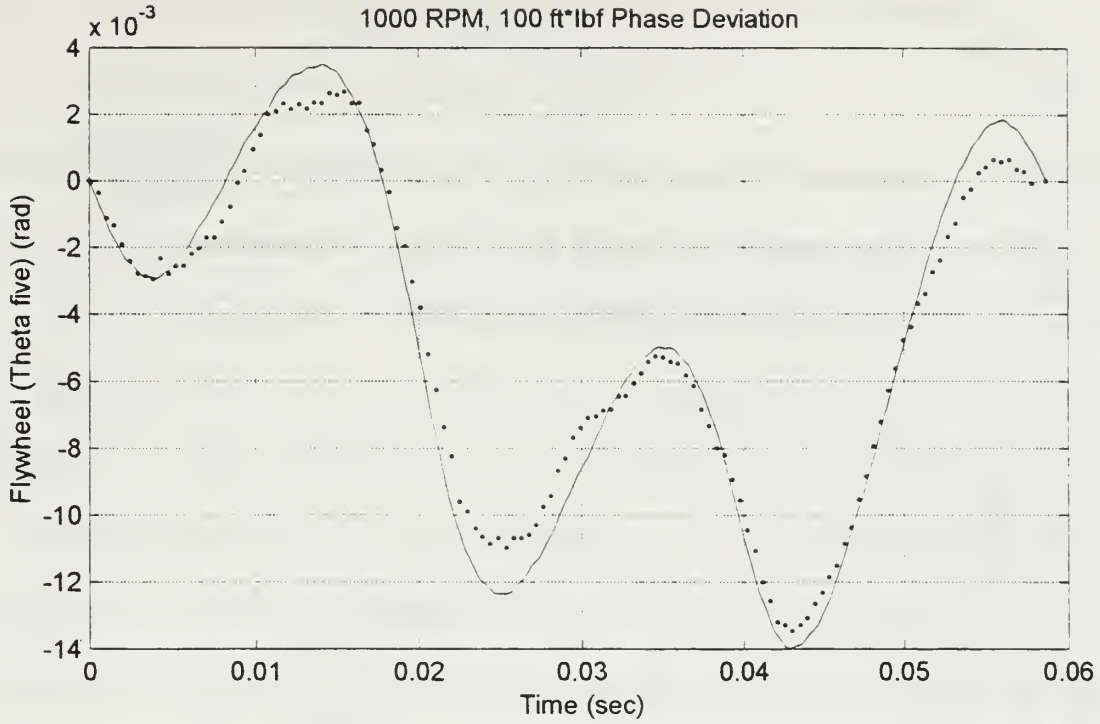
$$\varepsilon(t) = \theta(t) - \overline{\omega} t \tag{17}$$

where $\overline{\omega}$ is the mean rotational velocity. This is the same comparison plot used by Hudson [Ref 18]. For a shaft rotating at a steady angular velocity, the phase deviation would be zero. The measured phase deviation shows the crankshaft position advancing during the power stroke of each cylinder, then retreating during the subsequent compression of the next cylinder. A comparison of the measured and predicted phase deviation is shown for the crankshaft nose and the flywheel (Figure 16). Plots for other runs are found in Appendix D. The phase deviation plot is particularly sensitive to assumed values of friction and load torque, and is also useful for validating the model inertias. For the 1000 RPM 100 Ft*lbf data, phase deviation shows a maximum error of about 9% at the peaks, which correspond to the cylinder compression strokes. Generally, the model predicted phase deviation is within 5% of the measured phase deviation for

both the flywheel and the crankshaft nose. The errors are calculated as a percentage of the maximum variation in phase deviation.

The Crankshaft Twist Plot is simply the value of θ_1-θ_5 as a function of time. It is a measure of the total twist along the entire length of the crankshaft. A comparison of measured and predicted crankshaft twist is shown in Figure 17. This plot is particularly useful for validating the magnitudes of the torsional rigidities used in the model. The peak values of the twist occur at the peaks of the gas torque, during the power stroke for each cylinder. As expected, the amount of twist is largest for cylinder #1, the farthest from the flywheel, and successively lower for cylinders #2 and #3. Correct values of torsional rigidity for the model should result in predicted twists comparable to the measured value. Comparison of the model predicted twist and the measured twist is shown in Figure 17, with additional plots in Appendix D. A max error of about 17% is seen at the peak twist values, corresponding to the cylinder power strokes. Generally, the model predicted twist is within about 5% of the measured twist.

As discussed in Appendix C, analysis of the crankshaft natural frequencies can also be used to validate the model parameters. The analytical values derived for the crankshaft inertias are considered to be reasonably accurate, so the only parameters to be adjusted are the torsional rigidities. These are set by the comparison of measured and predicted natural frequencies and modes as discussed in Appendix C. Further arbitrary adjustment of the parameters to correct the errors in the Phase Deviation and Crankshaft Twist plots is not supported.



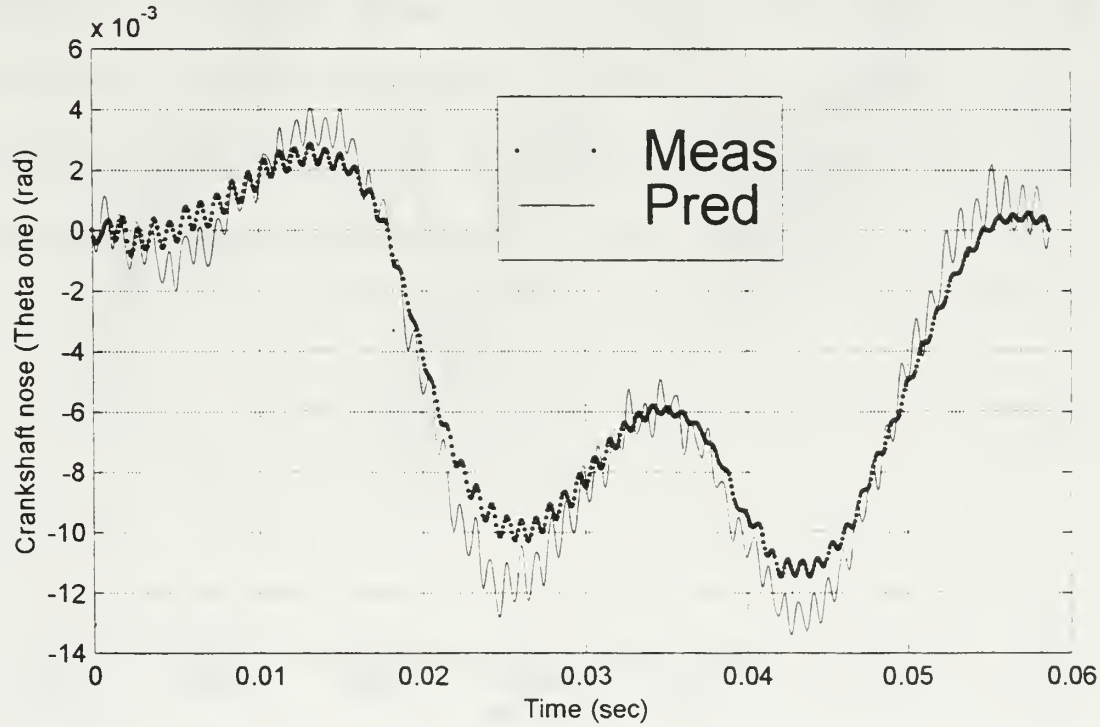


Figure 16. Phase Deviation (1000 RPM, 100 Ft*lbf)

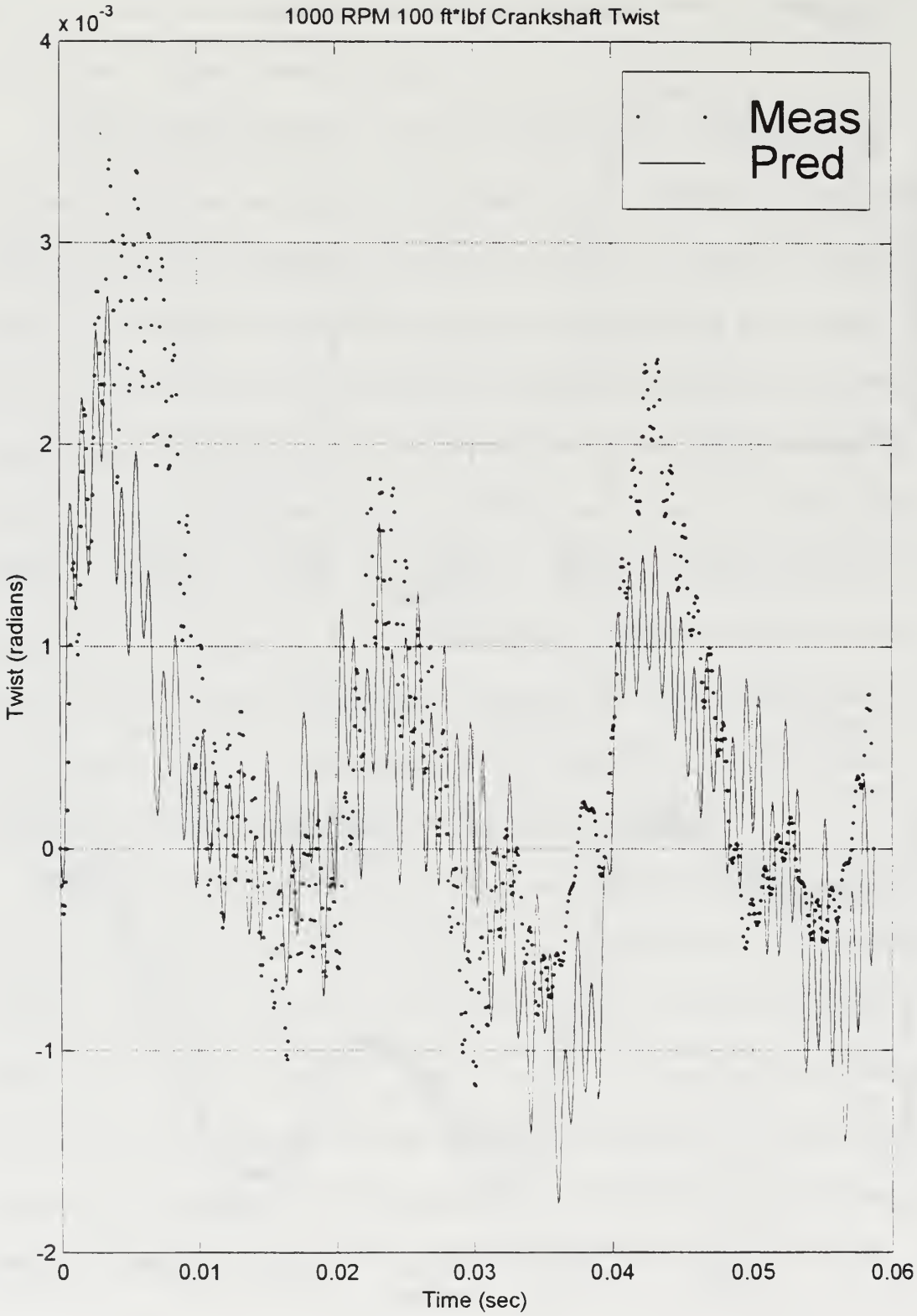


Figure 17. Crankshaft Twist (1000 RPM, 100 Ft*lbf)

B. COMPARISON OF PREDICTED AND MEASURED TORQUES

Inverse solution of the equations of motion allows prediction of the cylinder gas torques, given the time-resolved angular position at two points, θ_1 and θ_5. Comparison of measured and predicted gas torques for the individual cylinders is shown in Figure 18. Comparison is not good for individual torques. Besides quantitative errors of over 50% at certain points, the predicted torques show misplaced and inappropriate peaks. However, it appears that the errors for a particular predicted cylinder torque have corresponding offsetting errors in the predicted torque for the other cylinders. This is evident when measured and predicted total gas torques are compared (Figure 19). The predicted total gas torque shows peak-to-peak errors of less than 5%, plus a phase lag of about 5-15 degrees. Some of this error is from the numerical solution methods, as discussed in the previous chapter. This agreement is good enough to be used for localized fault detection. For this particular run of the engine, cylinder #1 gas pressure is significantly lower than the other two cylinders, and the predicted results detect this anomaly. Plots for the other data runs are contained in Appendix D.

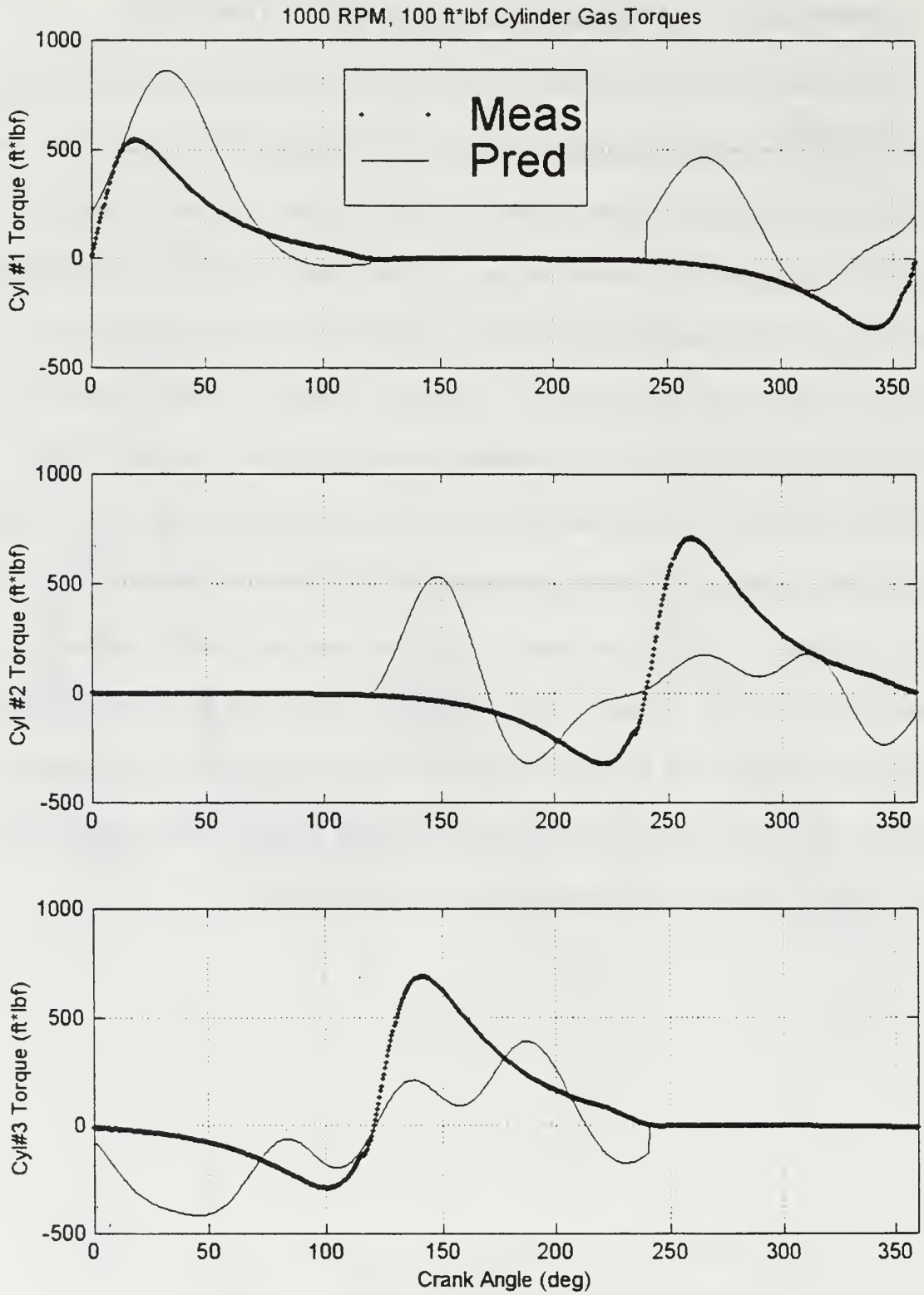


Figure 18. Individual Cylinder Gas Torques (1000 RPM, 100 Ft*lbf)

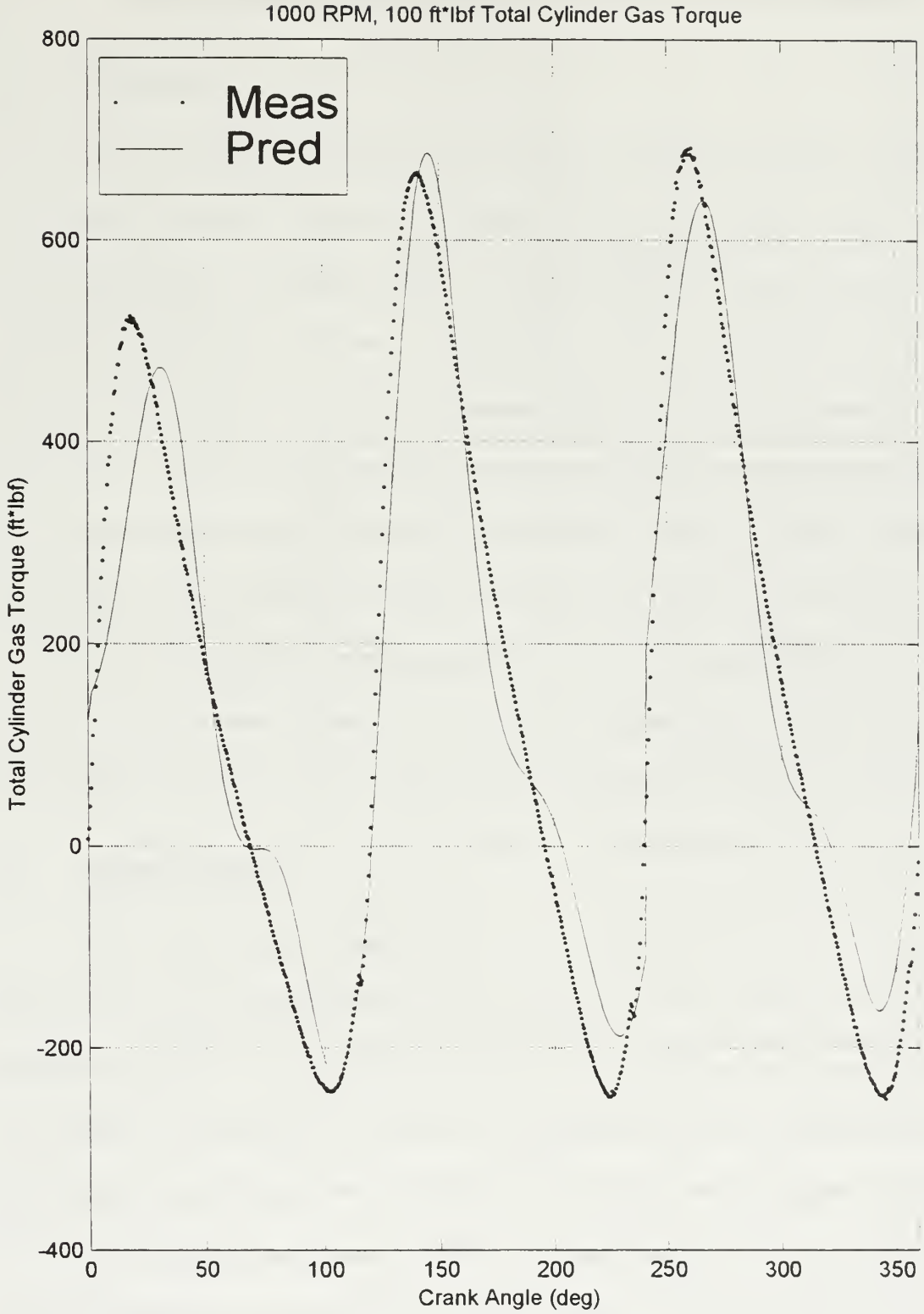


Figure 19. Total Gas Torque (1000 RPM, 100 Ft*lbf)

VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

A. SUMMARY

A three cylinder, two stroke Diesel engine was instrumented with a proximeter and an optical encoder for time-resolved angular position measurement at the flywheel and crankshaft nose. A torsional model for the engine crankshaft was developed and the corresponding equations of motion were formulated. Two separate numerical solution methods were developed to solve for the angular positions, given the measured cylinder gas torques. These methods were used to calibrate the parameters of the torsional model. An inverse solution method was devised to determine the cylinder gas torques, given the time-resolved angular positions at two of the degrees of freedom; the flywheel and the crankshaft nose. This inverse solution method was shown to be applicable for two-stroke engines of three or fewer cylinders, or for four-stroke engines of four or fewer cylinders. The predicted cylinder gas torques were compared to measured cylinder gas torques.

B. CONCLUSIONS

The torsional model accurately describes the dynamics of the actual crankshaft. Experimental data demonstrated that the model correctly predicted phase deviation at the crankshaft endpoints with an error of less than 5%. The model predicted crankshaft twist with an error of less than 20%. Predicted natural frequencies from the model agreed with the measured frequency spectrum to within 5 Hz for the three vibration modes observed.

The Finite Element Method (FEM) for direct integration of the equations of motion agreed with the Time-marching ODE method to within 1%, and it reduced

computational time by a factor of 100 because no iterations were necessary. It was used as the primary means for direct integration of the equations of motion in this study.

The inverse method for predicting cylinder gas torques showed significant errors in predicted individual cylinder gas torques. Quantitative errors of over 50%, as well as significant wave shape errors, make this method inadequate for reliable prediction of individual cylinder torques. However, it is the author's opinion that this error originates in the numerical method used. Specifically, signal filtering tends to create errors in the endpoints for the representative cycle. Further analysis of this problem may result in successful prediction of individual cylinder torques.

The inverse method is successful in predicting total cylinder gas torque. Predicted total gas torque errors were less than 5%, with slight phase lag errors of 5-15 degrees. The predicted total gas torque successfully detected a low pressure in cylinder #1, showing that the method is capable of localizing certain faults to a particular cylinder.

C. RECOMMENDATIONS

The failure of the inverse method to predict individual cylinder torques is most likely due to problems with the numerical solution methods. The FFT signal filtering induces some errors, which were not sufficiently corrected. The engine speed was not exactly steady during data collection, so the filtering process backs out a monotonic trend in the phase deviation. That is, the filtered phase deviation is no longer exactly cyclic. A correction of some sort should be made for this linear error. The process of signal filtering also tends to alter the endpoints of the representative cycle, inducing significant errors in the calculated torques at the endpoints. An alternative signal filtering process, which avoids these errors, might correct the errors in the results.

The assumption of constant parasitic force to model piston ring friction is not validated. The presence of crank-angle specific friction "sticking" points would have a significant effect on the results. One solution would be to use a motoring dynamometer on the engine to produce a graph of the engine friction as a function of crank angle. Also, a more detailed analysis of theoretical piston ring friction could lead to more accurate modeling.

From measurements obtained in this study, there is an unknown fault causing cylinder #1 to have a lower gas pressure than the other two cylinders. As a first step, the fuel injectors for cylinders #1 and #2 should be swapped in order to determine the cause of the low pressure in cylinder #1. For follow-on experimentation, data should be collected for engine runs with known faults. For instance, a defective fuel injector could be installed to test the method's ability to detect a specific fault.

The use of angular speed measurement internal to the engine would expand this method to engines with more cylinders. Although this would be a difficult process for an existing engine, mass-produced engines might have such an internal detector installed for relatively little extra cost.

The method for determining TDC for cylinder #1 is inadequate. Currently, the procedure of Appendix F, Ref [18] is used to orient the TDC signal on the encoder to TDC for the engine. But TDC for the engine is established by a mark inscribed on the crankcase and the forward counterweight on the cam follower shaft. Although this shaft is directly geared to the crankshaft, gear backlash results in an error of one or two degrees when the engine is rotated to TDC. This small error has a significant effect on the

magnitudes of the cylinder indicated torques. A better method would be a mark inscribed on the flywheel and shroud to ensure correct TDC alignment.

A sprocket and proximeter assembly might be a more useful means of collecting data at the front end of the crankshaft. A 42 tooth sprocket has been obtained which may be mounted on the crankshaft nose with the pulley mounting bolt. Since the number of teeth on the flywheel (126) is a whole number multiple of 42, a precise alignment could be made to calibrate the static phase difference between the two ends of the crankshaft to zero. Then the instantaneous twist of the crankshaft could be very accurately measured during operation, similar to the method described by Mauer and Watts [Ref 2]. Additionally, this data collection method would eliminate the natural frequency torsional vibrations of the optical encoder and flexible coupling.

APPENDIX A. GEOMETRY OF ROTATING COMPONENTS

While a mass-elastic model has already been developed for this crankshaft [Ref 23], it is necessary to independently calculate the model parameters in order to verify them, and to account for differences in the specific engine used in the research. This analysis was carried out using methods and equations from Wilson [Ref 24]. The descriptions and values for certain constants used in subsequent equations are listed in Table 6.

Table 6. Equation Constants

Symbol	Description	Value
D_{cp}	Diameter, crankpin	2.50 in
D_{jb}	Diameter, journal bearing	3.00 in
g	Gravitational Acceleration	386 in/sec^2
K_{gyr}	Radius of Gyration	
L_e	Length, effective	
L_{cp}	Length, crankpin	1.60 in
L_{jb}	Length, journal bearing	1.50 in
R	Crankpin eccentricity	2.25 in
R_i	Counterweight Inner Radius	1.80 in
R_o	Counterweight Outer Radius	4.02 in
T_{ct}	Counterweight Thickness	0.83 in
T_{wb}	Crankweb Thickness	1.00 in
W	Weight	
W_{wb}	Crankweb width	3.84 in
ρ	Specific weight of steel	0.283 lbf/in^3

1. Rotating Inertia

An arbitrary objects mass polar moment of inertia is calculated as:

$$J = \frac{WK_{gyr}^2}{g} \tag{A1}$$

where W is the weight of the object, K_{gyr} is the radius of gyration, and g is acceleration due to gravity.

The polar moment of inertia for a main journal bearing is determined as for a solid circular cylinder whose axis is aligned along the axis of rotation:

$$J_{jb} = \frac{WK_{gyr}^2}{g} = \frac{\pi D_{jb}^4 L_{jb} \rho}{32g} \tag{A2}$$

Crankpins are also determined as a solid circular cylinder, but the cylinder axis is parallel to and offset from the axis of rotation by the eccentricity:

$$J_{cp} = \frac{WK_{gyr}^2}{g} = \frac{\pi}{4g} D_{cp}^2 L_{cp} \rho \left[\frac{D_{cp}^2}{8} + R^2 \right] \tag{A3}$$

As seen in Figure 20, the Crankwebs are of three distinctive types. Crankwebs of type (a) connect crankpins for cylinders 1 and 3 to the outer journal bearings, and crankwebs of type (b) connect those same crankpins to the inner journal bearings. Crankwebs of type (c) connect the crankpin for cylinder 2 to the inner journal bearings.

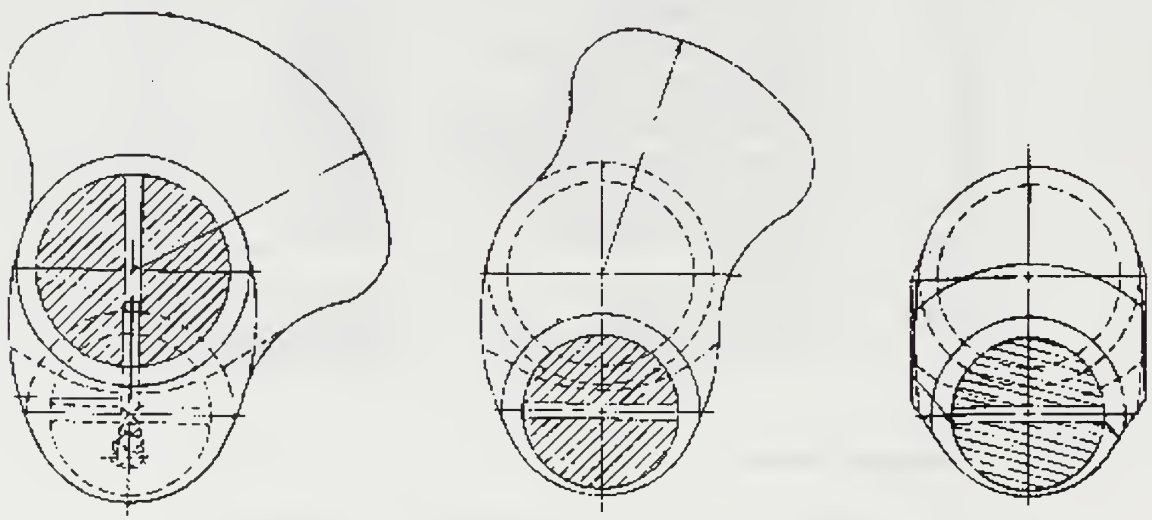


Figure 20. Crankweb Forms. From Ref [25]

For all three types, the core geometric shape is an ellipse with one focus centered on the journal bearing and the other focus centered on the crankpin. From Table 10.13 of Wilson [Ref 24], the radius of gyration for an ellipse can be calculated as:

$$K_{ellipse} = \frac{(a^2 + b^2)}{16} + c^2 \tag{A4}$$

where a and b are the minor and major axis, respectively, and c is the offset of the center of gravity from the axis of rotation. J can then be calculated for the elliptical portion from equation (A1). The counterweight lobes are then treated as semi-circular segments, and their contribution to mass polar moment of inertia is:

$$J = \frac{\pi \rho T_{ct} \left(\alpha / 360\right)}{2g} \left(R_o^4 - R_i^4\right) \tag{A5}$$

where α is the angle subtended by the counterweight lobe; 120° for crankweb (a) and 70° for crankweb (b).

Rotating inertia for the dynamometer is taken from Ref [26]. The coupling shaft between the flywheel and the dynamometer is calculated as a circular cylinder using Equation (A1).

2. Torsional Rigidity

For modeling of the torsional rigidity, the components of the crankshaft must be mathematically converted to an equivalent shaft of a constant diameter. This is a simple geometric problem for static twisting of the crankshaft, but becomes very complex when considering the dynamic crankshaft twist during engine operation.

Wilson [Ref 24] presents a derivation of the torsional rigidity for various crankshaft components. The rigidity for a solid cylindrical shaft of diameter D is:

$$K = \frac{\pi D^4 G}{32 L_e} \tag{A6}$$

This equation can be used to determine the torsional rigidity of any solid cylindrical component, including the journal bearings and crankpins. For the crankweb, an effective

length is derived based on the bending theory of beams. The crankweb is treated as a beam subjected to a bending force by the torsion on the crankshaft:

$$L_{ecw} = D^4\left[\frac{0.942R}{T_{wb}W_{wb}^3}\right] \tag{A7}$$

The equivalent shafting between concentrated inertia points in the model can be determined by adding up the torsional rigidity of their components. However, the equations above assume an unconstrained crankshaft deflection due to an applied static torque. The true equivalent length of the crankshaft elements will be modified by several other factors: local deformation where the journal bearings and crankpins join the crankweb, bearing restraint on the journal bearing, and non-ideal lever arm at the crankweb because the bearing and crankpin are not attached at a single point. [Ref 24]

Two empirical relations are considered in this study to account for increased rigidity of the crankshaft elements due to dynamic constrained operation while mounted in the engine block. The first was devised by Carter [Ref 27]:

$$L_{ejb} = D^4\left[\frac{L_{jb}+0.8T}{D_{jb}^4}\right] \quad L_{ecp} = D^4\left[\frac{0.75L_{cp}}{D_{cp}^4}\right] \quad L_{ecw} = D^4\left[\frac{1.5R}{T_{wb}W_{wb}^3}\right] \tag{A8}$$

and the second by Wilson [Ref 24]:

$$L_{ejb} = D^4\left[\frac{L_{jb}+0.4D_{jb}}{D_{jb}^4}\right] \quad L_{ecp} = D^4\left[\frac{L_{cp}+0.4D_{cp}}{D_{cp}^4}\right] \quad L_{ecw} = D^4\left[\frac{R-0.2(D_{jb}+D_{cp})}{T_{wb}W_{wb}^3}\right] \tag{A9}$$

Table 7 compares the results obtained by each of the methods so far described with the values from the model supplied by Detroit Diesel [Ref 23]. The general assumptions for the two empirical methods are clear: Carter's method assumes a stiffer crankpin and more flexible crankwebs while Wilson's method is reversed [Ref 24]. The determination of

torsional rigidities for the model is not straightforward; it will require some calibration from measured data.

Table 7. Calculated Torsional Rigidities

	Torsional Rigidity (10^6 lbf*in/rad) by method:			
Value	Unconstrained	Carter	Wilson	Model [Ref 23]
K_{jb}	7.07	41.5	35.3	
K_{cp}	4.60	38.4	17.7	
K_{cw}	31.5	19.8	49.4	
K_1	1.94	3.11	3.22	3.47
K_2, K_3	2.37	6.61	7.98	9.55
K_4	3.49	10.8	12.15	13.50

3. Auxiliary Loads

The auxiliary loads, with the exception of the oil pump, are driven off the timing gear, just forward of the flywheel. For the model, the total of the auxiliary load torque is considered to be placed at θ_5. The contribution of the individual loads can be determined by deriving the power required, then the torque is related to the power P by the equation:

$$T = \frac{P}{2\pi N} \tag{A10}$$

Two cam shafts are driven off the timing gear, at the rear of the crankshaft and just forward of the flywheel. The load due to these camshafts primarily results from three components: the bearing friction, the operation of the fuel injector pistons, and the compression of springs associated with the injectors and the valves. Bearing friction is determined in the next section. The action of the injector pistons is considered as a polytropic isothermal process of compression from 50 to 2800 psig. Work required to compress the springs is calculated from:

$$W_{spring} = \frac{1}{2} K_{spring} (\delta_2^2 - \delta_1^2) \tag{A11}$$

where δ_1 and δ_2 are the initial and final spring deflection, respectively. The load torque for the cam shafts is calculated from the work done by the shafts over one rotation.

The fuel pump is a positive displacement gear type pump that provides a flowrate Q of 1 gpm at 65 psi when the engine is at 2800 RPM. Assuming a pressure at the input of about 15 psi, the torque required can be determined from the power P:

$$P = Q\Delta p \tag{A12}$$

Since the flowrate Q is proportional to the speed N, the fuel pump torque will be a constant value regardless of engine speed.

The water pump is a centrifugal pump that provides 37 gpm at 2800 RPM. Since a pressure drop was not provided from the service manual, its effect is estimated as comparable to the other auxiliary components.

A roots blower provides the scavenging pressure that clears the pistons at the bottom of the stroke. This blower is rated to provide 338 cfm at 2800 RPM.

The oil pump is a rotary style positive displacement pump rated to deliver 15 gpm at 2800 RPM. The power can be calculated from

$$P = \dot{m}h_A = (Q\rho)\left(\frac{\Delta p g_c}{\rho}\right) \tag{A13}$$

where h_A is the head provided by the pump $\dot{m}$ is the mass flowrate of the oil, ρ is the oil density, and Δp is the pressure increase. The increase in fluid head due to velocity increase is neglected.

4. Friction Losses

Journal bearing friction is accounted for by assuming that the two surfaces are completely separated by the lubricating film; that hydrodynamic lubrication is dominant.

Using the relations provided by Heywood [Ref 28], an equation to calculate the parasitic torque due to a bearing's friction is derived:

$$T_{fr} = fW_f r \Rightarrow f = \frac{2T_{fr}}{W_f D_b} = \frac{\pi^2 D_b \mu N}{\bar{h}\sigma}; \sigma = \frac{W_f}{L_b D_b} \tag{A14}$$

where f is the friction factor, W_f is the bearing load, μ is the oil viscosity, $\bar{h}$ is the average bearing clearance, N is the rotation speed in RPM, and L_b and D_b are the bearing length and diameter. This can be solved for torque to yield:

$$T_{fr} = \frac{\pi D_b^3 L_b \mu}{4\bar{h}}\dot{\theta} \tag{A15}$$

The friction torque is proportional to rotation speed, and independent of bearing load under the assumption of hydrodynamic lubrication. This equation can be applied to the crankshaft main bearings, the crankpins, and the camshaft bearings. There are four main bearings, the coefficients C_1, C_2, and C_3 in the equations of motion are determined as one-third of the total friction for the main bearings. Crankpin friction is lumped in with the piston ring friction, and the friction due to the camshaft bearings is included with the auxiliary loads.

Piston ring friction is not easily modeled analytically, but is instead estimated by empirical methods. From Heywood [Ref 28], piston friction is considered as the sum of two components: a boundary friction generally proportional to engine loading, and a hydrodynamic friction proportional to piston speed. The exact relation is not only difficult to predict for any one engine, it will change as the engine's condition varies. A more detailed study of the ring friction is beyond the scope of this study. Although it is expected that the amount of piston friction varies as a function of the piston speed, a constant parasitic force is assumed, which corresponds to a parasitic torque T_{par} which

varies with crank angle, as in Equation (A16). The derivation of the force/torque relationship is detailed in the next Appendix. Since the magnitude of T_{par} is small relative to T_{cyl}, this is considered a reasonable approximation.

$$T_{par} = F_{par} R \left| \frac{\sin(\theta + \phi)}{\cos\phi} \right| \tag{A16}$$

The estimates for loads and frictions formulated above are meant to provide a relative relation between them. For this study, actual losses were determined from the measured pressure data (Table 8). The values for T_{load} and T_{par} were each estimated as a fraction of the total losses.

Table 8. Empirical Friction Losses

Engine Speed (RPM)	Dyno Load (ft*lbf)	Cyl #1 Average Gas Torque	Cyl #2 Average Gas Torque	Cyl #3 Average Gas Torque	Total Average Gas Torque	Total Losses	Efficiency
1000	80	24.9	51.5	56.1	132.5	52.5	60.4 %
1000	100	32.4	58.7	62.5	153.6	53.6	65.1 %
1000	135	44.9	69.2	73.1	187.2	52.2	72.1 %
1500	135	53.7	79.0	82.4	215.1	80.1	62.8 %
1500	160	63.2	87.9	91.9	243.0	83.0	65.8 %
2000	160	72.7	103.3	102.6	278.6	118.6	57.4 %

APPENDIX B. GEOMETRY OF RECIPROCATING COMPONENTS

The nonlinear motion of the piston and connecting rod (Figure 21) presents unique geometrical and mathematical problems when modeling a reciprocating engine. The following derivations are common in the literature, but are presented here to document the detailed analysis required to formulate the torsional model.

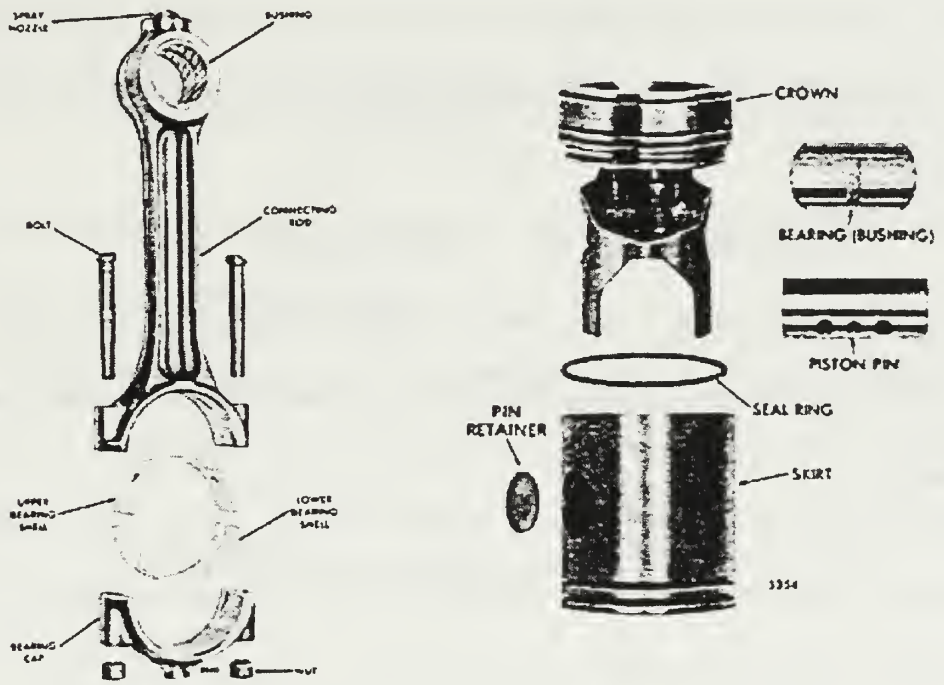


Figure 21. Piston and Connecting Rod. From Ref [19]

1. Indicated Cylinder Torque

There are several ways to derive the relation between the gas pressure present in the cylinder and the resulting indicated torque applied to the crankshaft. Piston ring friction is ignored for this derivation; it is corrected for by a parasitic force as described in the previous Appendix. Figure 22 shows the relation of the piston to the crankshaft. A static analysis assumes two forces applied at the piston pin. F_p is the net force due to the

indicated cylinder pressure, minus the reference pressure applied to the underside of the piston from the crankcase ($F_p = P_{net}A_p$). Additionally, a reaction force F_r is applied by the cylinder walls on the piston rings; this force constrains the piston to linear motion in the cylinder. The resultant force in the direction of the connecting rod is:

$$F_{cr} = \frac{A_p\left(P_{cyl} - P_{ref}\right)}{\cos\phi} \tag{B1}$$

where A_p is the cross-sectional area of the piston, P_{cyl} is the indicated cylinder pressure, and P_{ref} is the reference pressure. The crank angle θ and the connecting rod angles ϕ and γ are related by

$$L\sin(\phi) = R\sin(\theta) \qquad \text{and} \qquad \sin(\gamma) = \sin(\theta+\phi) \tag{B2}$$

The resultant torque applied at the crankshaft is then calculated by taking the cross-product of the connecting rod resultant force vector and the crankpin position vector, and simplifying:

$$T_{cyl} = \vec{F}_{cr} \times \vec{R} = \left|\vec{F}_{cr}\right|\left|\vec{R}\right| \sin\gamma = A_p\left(P_{cyl} - P_{ref}\right)R\frac{\sin(\theta+\phi)}{\cos\phi} \tag{B3}$$

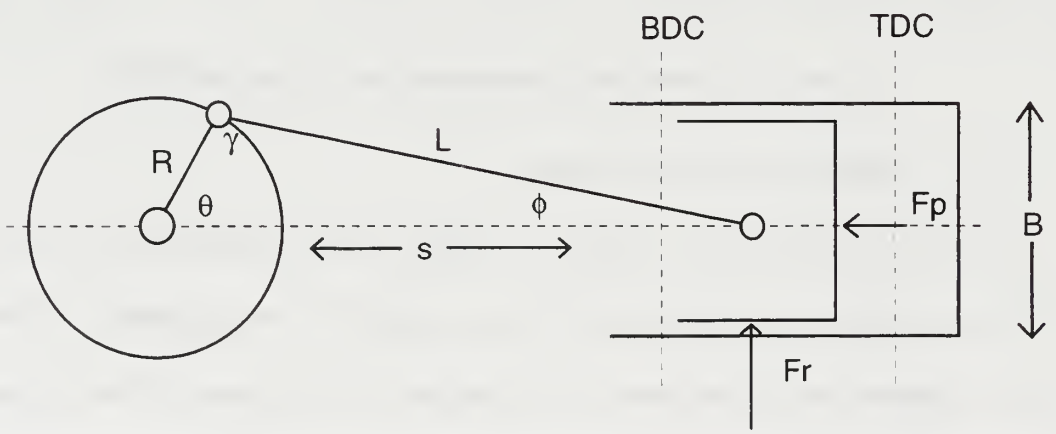


Figure 22. Geometry of Reciprocating Components

The same result is also achieved by equating the work done at the piston with the work done by the torque on the crankshaft, as demonstrated by Taylor [Ref 29]:

$$P_{net}dV = T_{cyl}d\theta \Rightarrow T_{cyl} = P_{net}A_p\frac{\dot{s}}{\dot{\theta}} \tag{B4}$$

Taking the relation for s from Wilson [Ref 24],

$$s = R\cos\theta + L\cos\phi = R\cos\theta + \left[L^2 - R^2\sin^2\theta\right]^{1/2} \tag{B5}$$

then

$$\dot{s} = \dot{\theta}\left[-R\sin\theta - \frac{R^2\sin\theta\cos\theta}{\left(L^2 - R^2\sin^2\theta\right)}\right] = \dot{\theta}\left[-R\sin\theta - \frac{R^2\sin\phi\cos\theta}{\cos\phi}\right] \tag{B6}$$

and using trigonometric relations, equation B4 will simplify to equation B3.

2. Reciprocating Torque

While the rotating parts of the crankshaft maintain a nearly constant angular velocity, the reciprocating components are alternately accelerated and decelerated in a constrained linear motion. At the crankshaft, this will be seen as a load torque while the piston is accelerated from TDC to its maximum speed, and will supply a torque as the piston is decelerated to BDC. Taylor [Ref 29] formulates a method of deriving this reciprocating torque.

First, it is necessary to determine the amount of the reciprocating mass. Clearly, the entire mass of the piston contributes, but only a portion of the connecting rod is reciprocating, and the rest must be considered rotating mass. A first approximation idealizes the connecting rod as two lumped masses connected by a massless shaft, with the same total mass and center of gravity as the real connecting rod (Figure 23). The center of gravity for the real connecting rod is simply found by balancing the rod; for this engine h = 3.5 in. and j = 5.3 in. The portion labeled W_1 is then added into the

reciprocating mass, while the remainder is considered part of the crankshaft rotating mass. Then, the instantaneous torque T_{rec} is found by equating the change in the reciprocating mass kinetic energy with the work done by the crankshaft:

$$d\left(\frac{W_{rec}}{2g}\dot{s}^2\right) = -T_{rec}\,d\theta \Rightarrow T_{rec} = -\frac{W_{rec}}{g}\ddot{s}\left(\frac{\dot{s}}{\dot{\theta}}\right) \tag{B7}$$

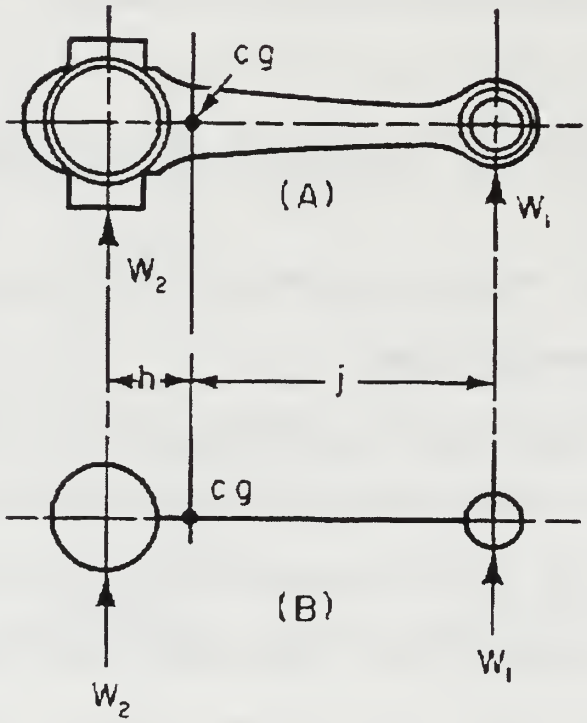


Figure 23. Idealized Connecting Rod. From Ref [29]

A small correction must then be made to account for the difference between the polar moment of inertia for the idealized connecting rod and the actual polar moment of inertia:

$$T_{rec,corr} = \left(J_{cr} - hj\frac{W_{cr}}{g}\right)\ddot{\phi}\frac{R\cos\theta}{L\cos\phi} \tag{B8}$$

where J_{cr} is the polar moment of inertia for the actual connecting rod. While Taylor derives series relations to state all values as functions of θ, for this study the angle

ϕ and the distance s are calculated directly, then differentiated numerically within the program.

3. Reciprocating Inertia

During engine operation, the reciprocating components contribute to the rotating inertia of the crankshaft. However, the magnitude of the reciprocating inertia varies as a function of the crank angle θ. The rotating motion of the crankshaft drives a linear motion of the piston in the cylinder. When the piston is at TDC, an incremental rotation of the crankshaft results in zero linear motion of the piston, while at 90° the same incremental rotation of the crankshaft results in maximum linear motion of the piston. Therefore, the influence of the piston mass on the inertia seen at the crankshaft will vary during crankshaft rotation. Normally, crankshaft inertial models include an average value of one-half the maximum reciprocating inertia to account for the reciprocating components. However, in this application, we are interested in crank-angle dependent values of torque, so we must account for this crank-angle dependent variation of reciprocating inertia. The reciprocating inertia can be calculated as a function of the reciprocating mass, the eccentricity, and the crank angle:

$$J_{rec} = \frac{W_{rec} R^2}{2g}(1 - \cos(2\theta)) \tag{B9}$$

The value W_{rec} is the weight of the reciprocating components, as determined previously.

Reciprocating inertia is a separate effect from the reciprocating torque already described. The changing value of reciprocating inertia accounts for the extra mass that, along with the rotating mass, must be accelerated when the crankshaft is accelerated. The reciprocating torque accounts for the energy required to accelerate this reciprocating

mass from zero to the current rotational speed of the crankshaft, before it is added to the rotating mass.

APPENDIX C. NATURAL FREQUENCY AND MODAL ANALYSIS

A modal analysis can be conducted for the crankshaft using the torsional model. Fourier analysis of the torsional vibrations measured at the optical encoder will show peaks corresponding to the measured natural frequencies. Comparison of the predicted natural frequencies from the model to these measured natural frequencies is a powerful calibration tool for fine-tuning the model.

The matrix equation describing the torsional mode of the crankshaft is repeated here:

$$[J]\ddot{\theta} + [C]\dot{\theta} + [K]\theta = [T] \tag{7}$$

Neglecting the damping effects, the natural frequencies are calculated by:

$$[K] - [J]\omega^2 = [0] \Rightarrow \omega^2 = eig\left\{[J]^{-1}[K]\right\} \tag{C1}$$

For the given model parameters (Table 1), the results are tabulated in Figure 24. There are six modes of natural vibration, with the first mode being the trivial rigid-body oscillation, where there is no crankshaft twist.

A natural vibration component due to the rigidity and inertia of the flexible coupling and the optical encoder disk is expected. The given values for the unit are:

$J_{rotor} = 1.45 \times 10^{-6}$ kg*m^2

$J_{coupling} = 3 \times 10^{-6}$ kg*m^2

$K_{coupling} = 200$ N*m/rad [Ref 21]

and given

$$\omega^2 = \frac{K_{coupling}}{(J_{coupling} + J_{rotor})} \tag{C2}$$

the natural frequency for torsional vibration for the coupling/encoder unit is:

$\omega = 1067$ Hz

Not surprisingly, a large response is seen in the measured data at precisely this frequency.

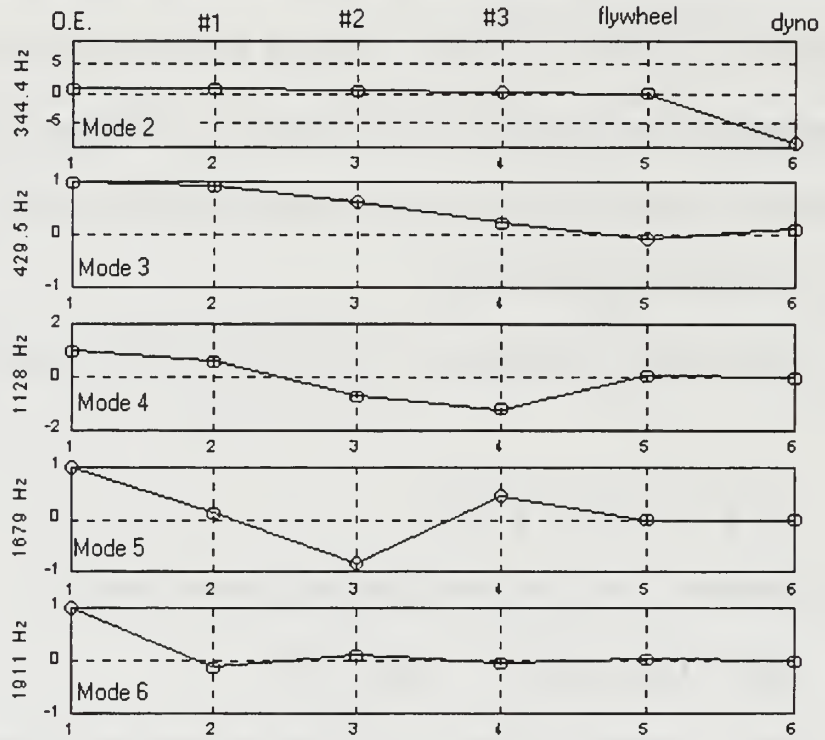


Figure 24. Torsional Vibration Modes

For highest resolution, a Fast Fourier Transform (FFT) is conducted on the measured angular velocity at the optical encoder over the entire 11 cycles of collected data. Figure 25 shows the frequency spectrum for measured data from the 1000 RPM, 100 ft*lbf run. Spectrums obtained for other engine speeds and loads are similar. Figure 26 is an expanded view of the low-frequency portion of the spectrum. A "comb" of amplitude spikes are seen, corresponding to harmonics of the engine rotation frequency, 16.7 Hz. As expected, a large frequency response is seen at about 1060 Hz, corresponding to the natural frequency of the encoder/coupling unit. Additional responses are seen at about 430 Hz, 1130 Hz, and 1910 Hz, and these agree with three of

the expected modes from the torsional model. A peak seen at about 2400 Hz is from an unknown source; it is not present in frequency spectrums taken at other engine speeds.

Natural frequencies that are predicted by the model but not seen in the spectrum are probably due to low amplitudes at the crankshaft nose. For instance, the expected 344 Hz frequency is mostly oscillation of the dynamometer rotor with respect to the flywheel; the crankshaft oscillation amplitude would be much smaller. As expected, each of the predicted modes has a node close to the flywheel because it contains the bulk of the system inertia. Measurements taken at the flywheel would be expected to show almost no high frequency vibration, and this is seen in the measured data.

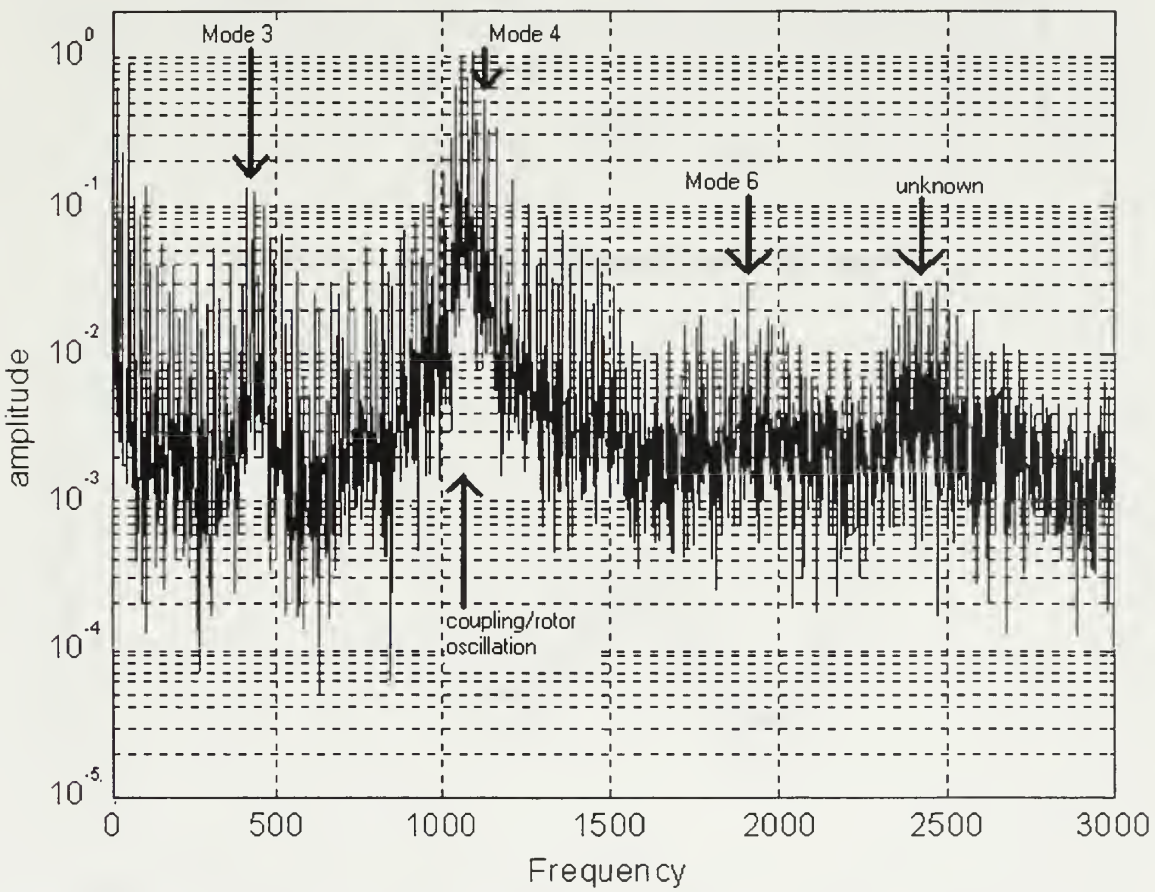


Figure 25. Frequency Spectrum for Measured Angular Velocity at θ_1 (0-3000 Hz)

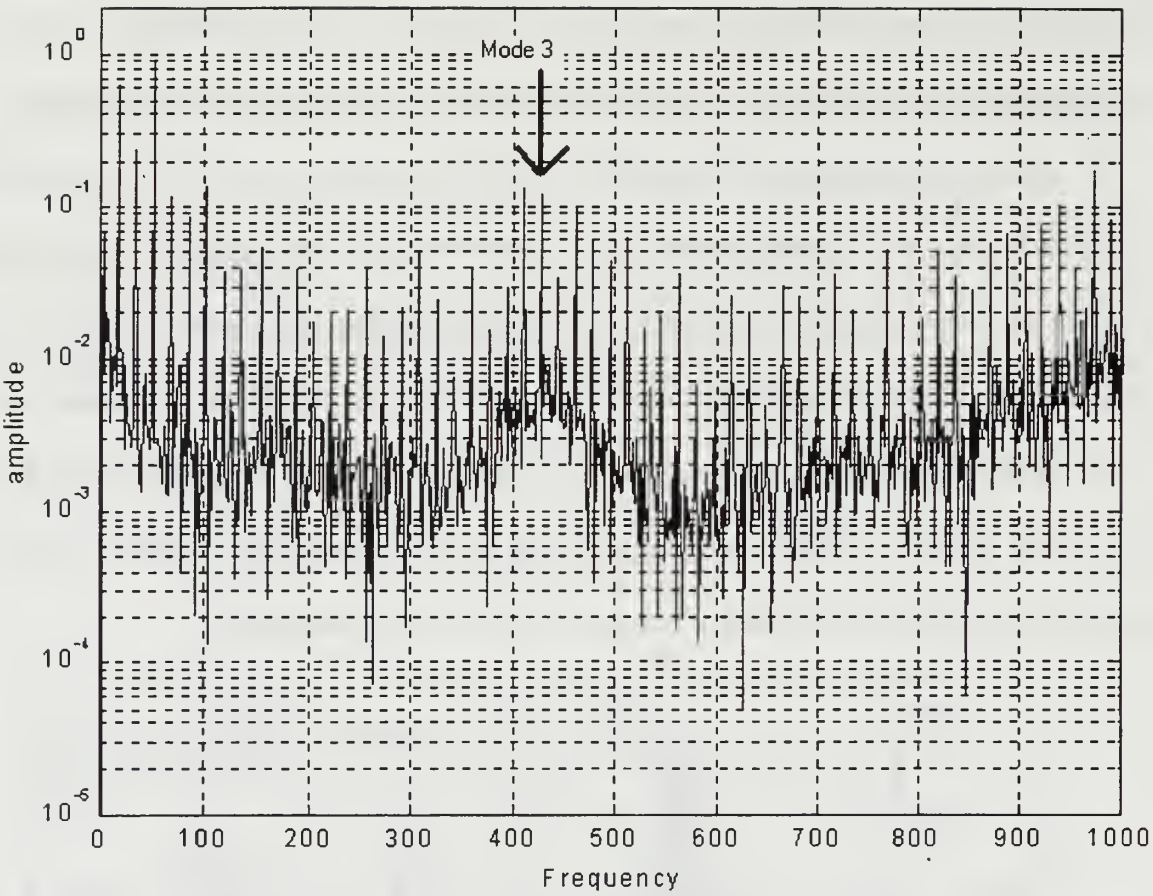


Figure 26. Low Frequency Spectrum (0-1000 Hz)

APPENDIX D. ADDITIONAL DATA PLOTS

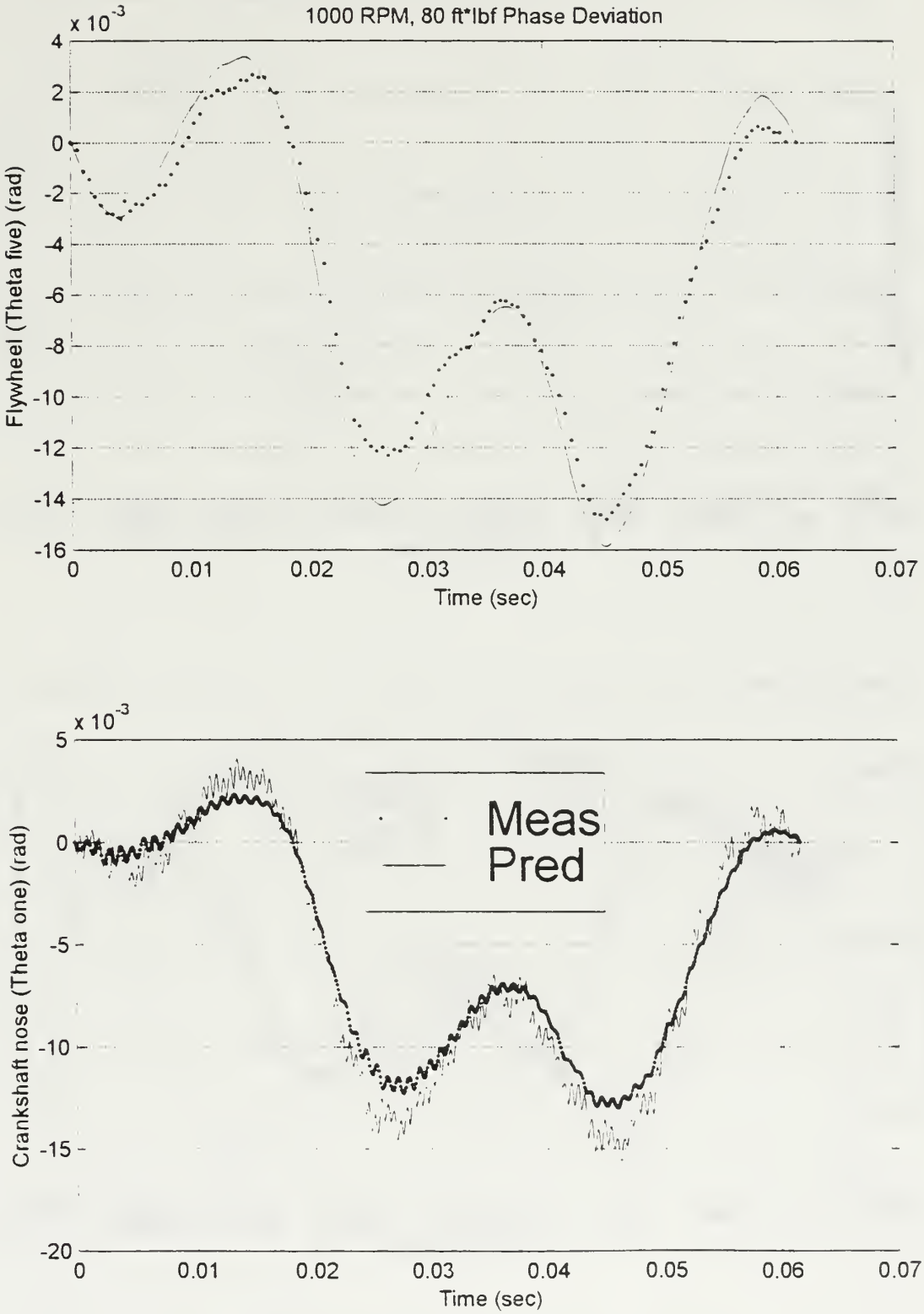
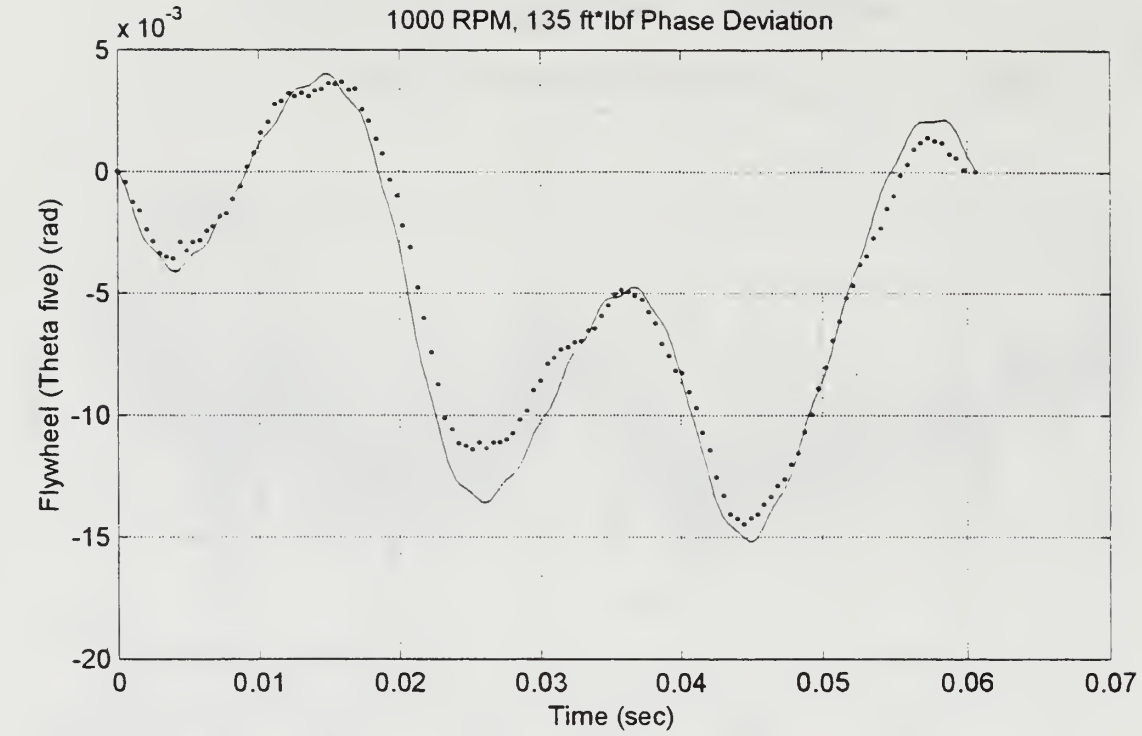


Figure 27. Phase Deviation (1000 RPM, 80 Ft*lbf)



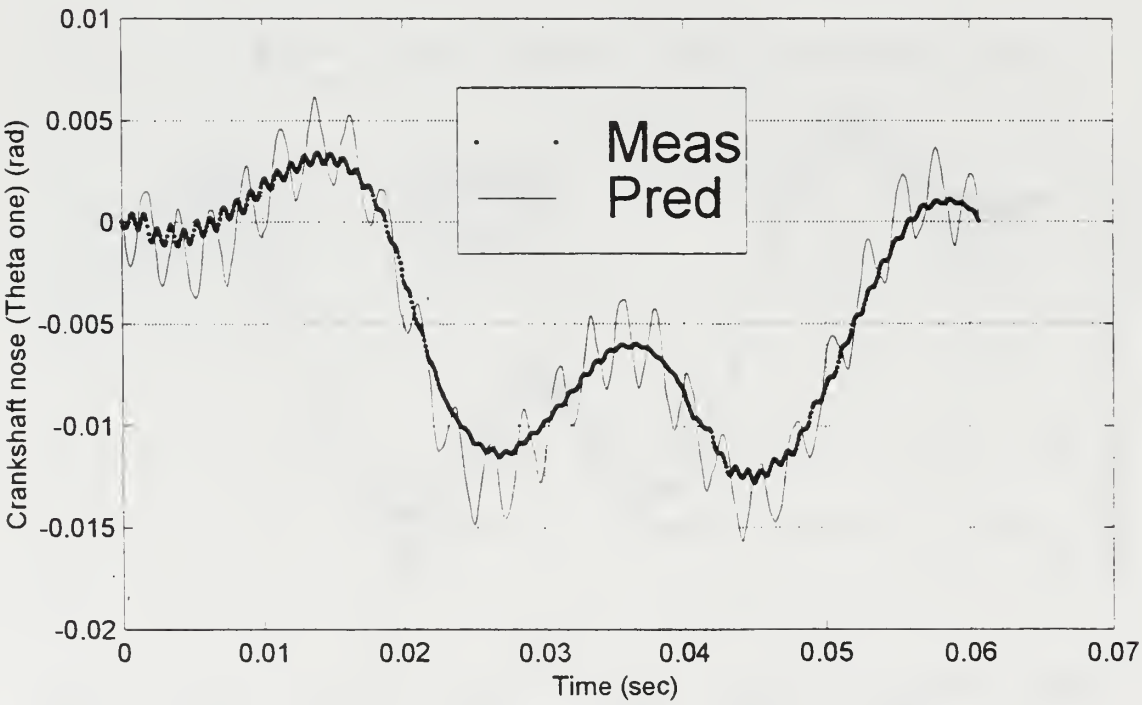


Figure 28. Phase Deviation (1000 RPM, 135 Ft*lbf)

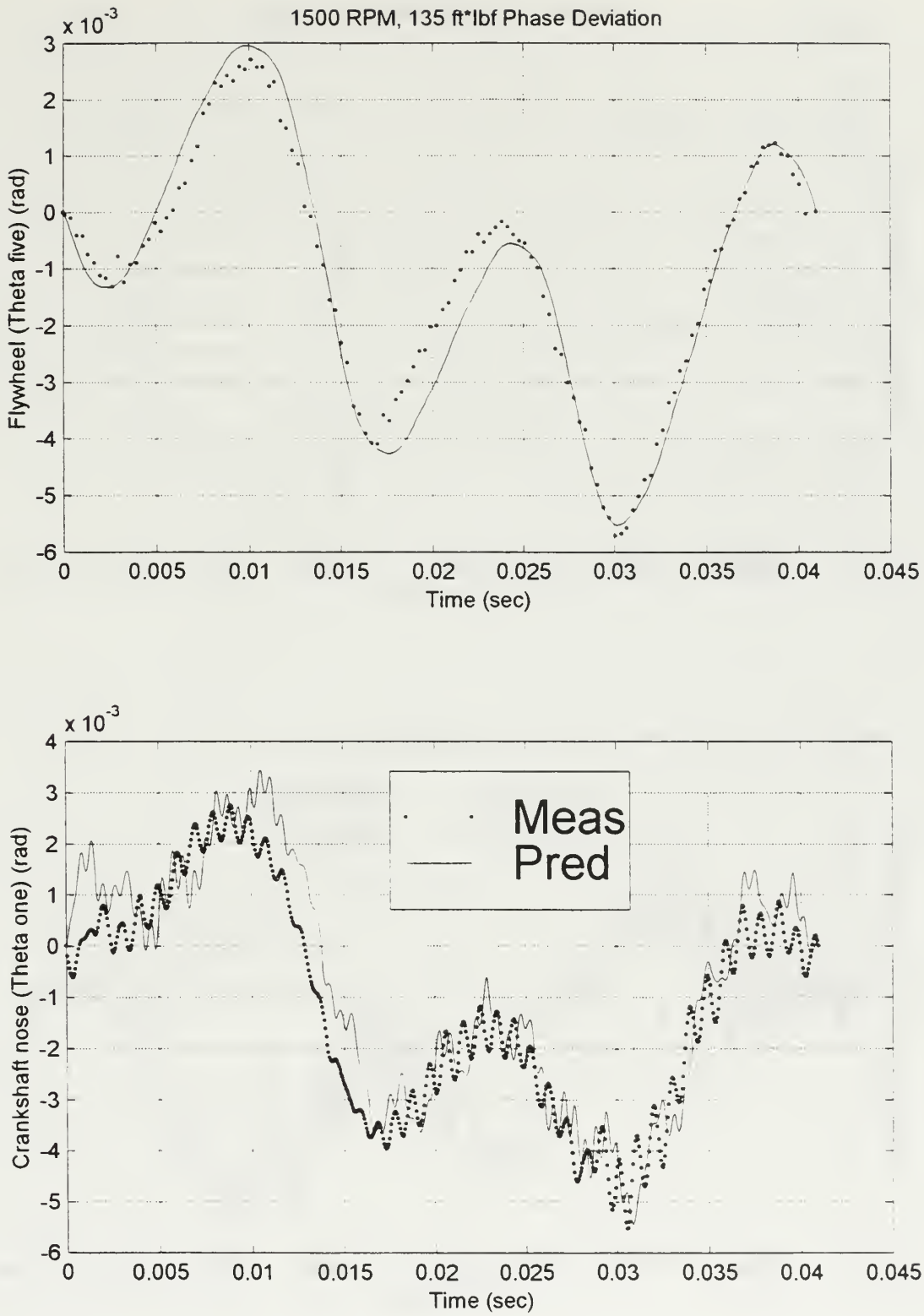
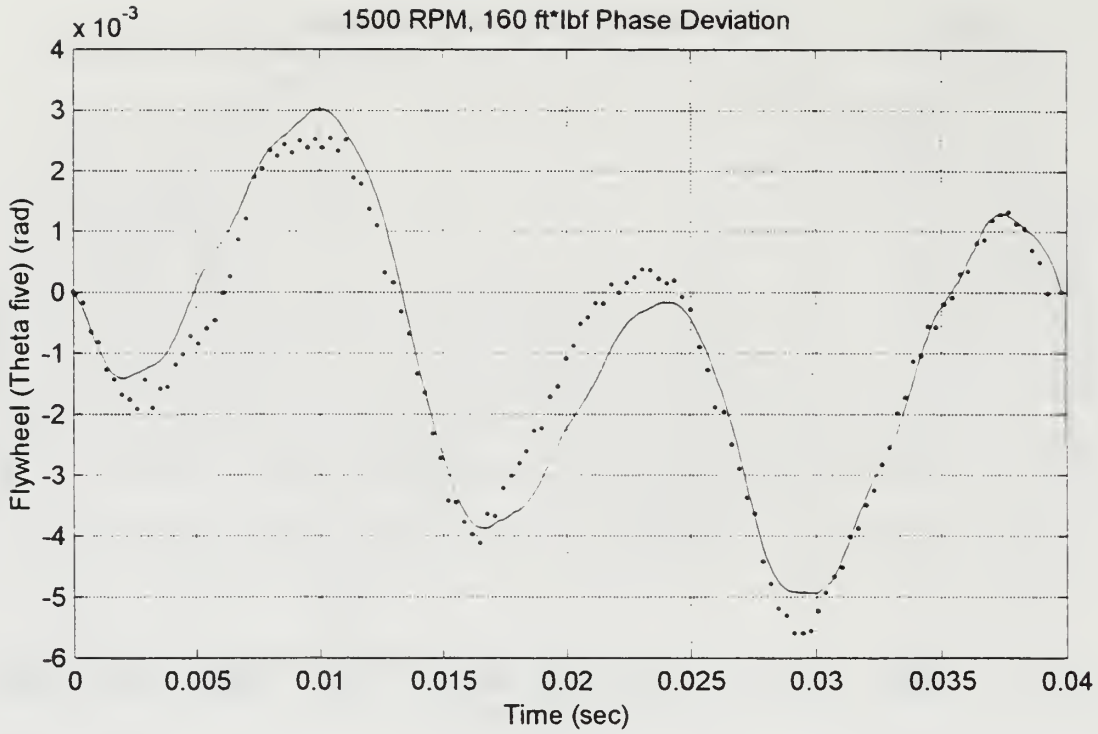


Figure 29. Phase Deviation (1500 RPM, 135 Ft*lbf)



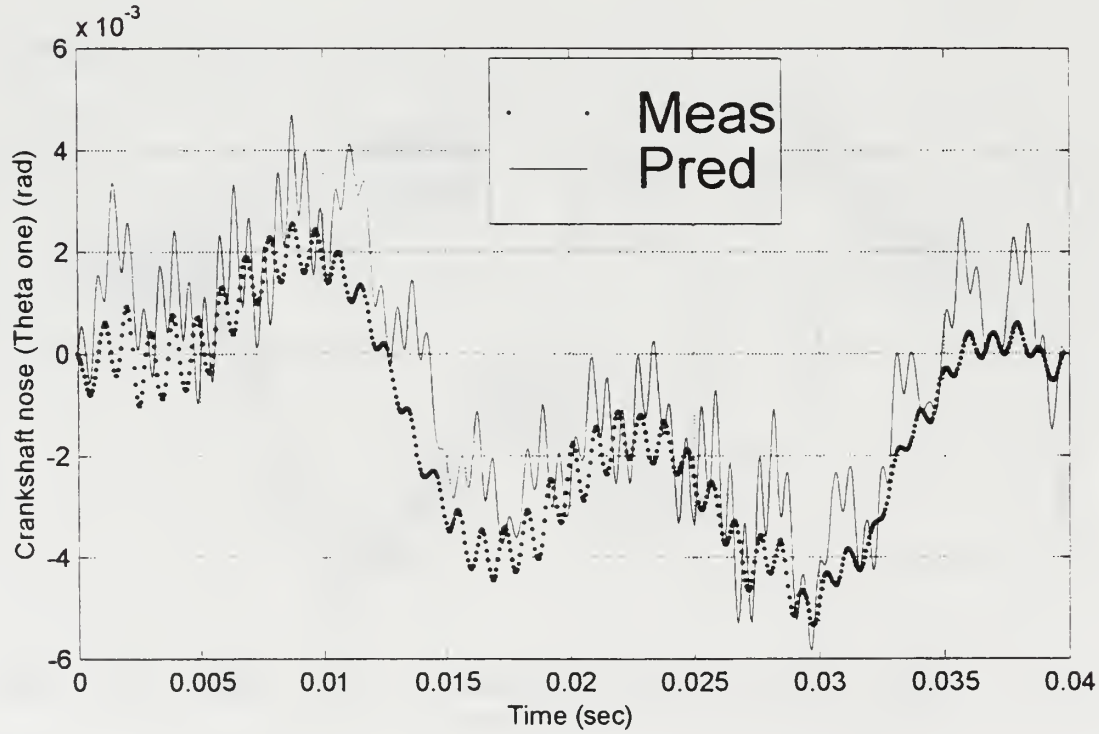


Figure 30. Phase Deviation (1500 RPM, 160 Ft*lbf)

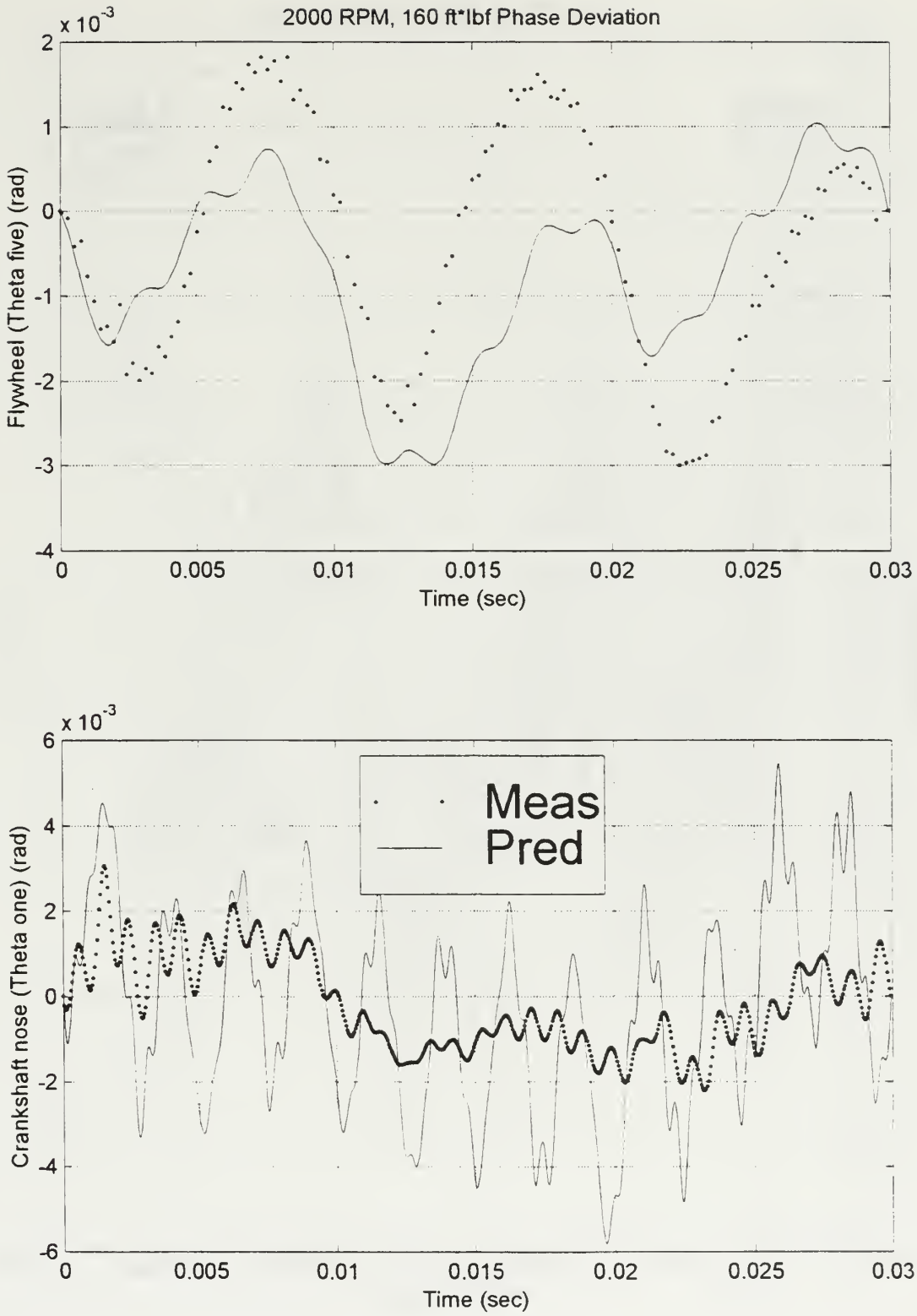


Figure 31. Phase Deviation (2000 RPM, 160 Ft*lbf)

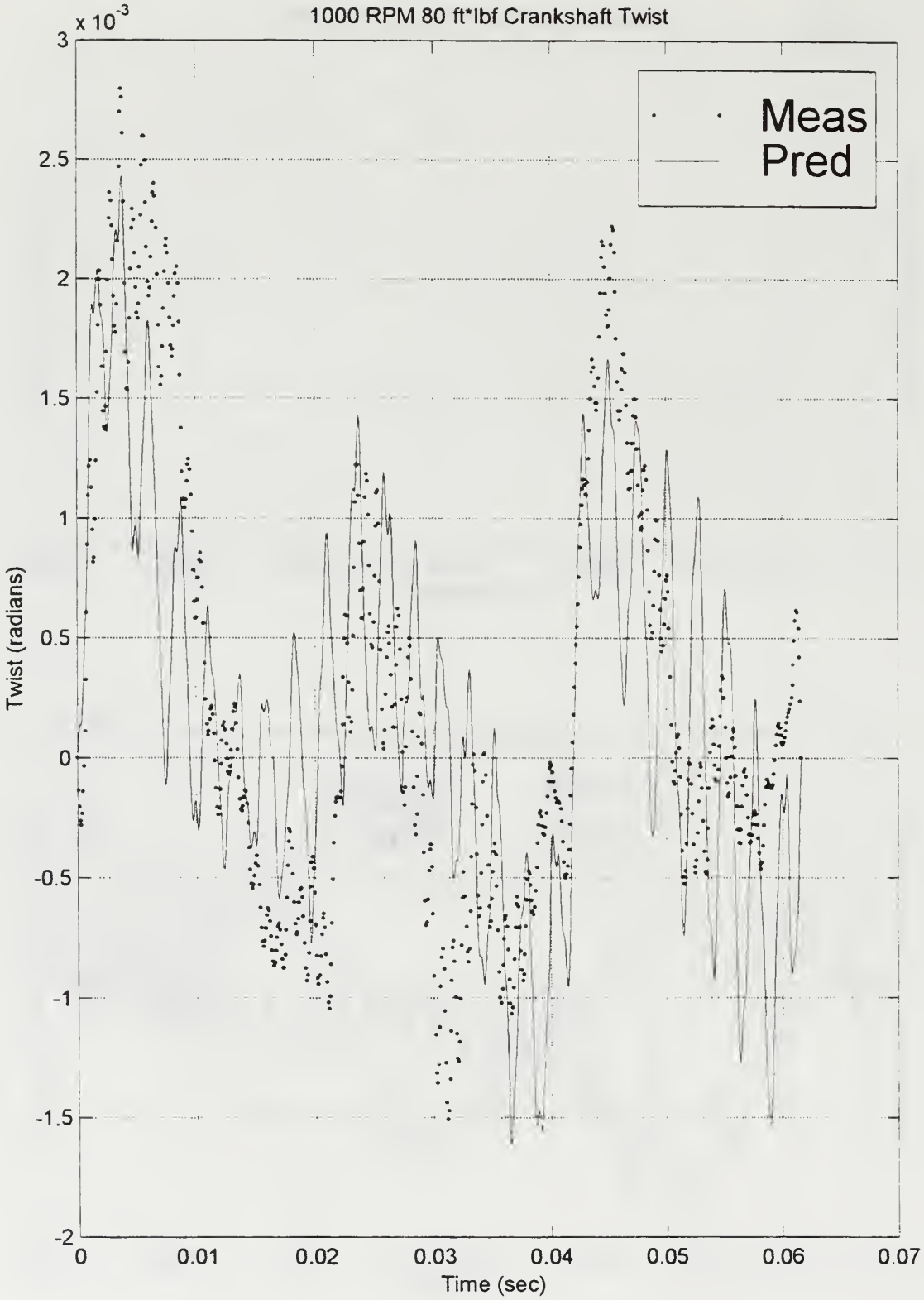


Figure 32. Crankshaft Twist (1000 RPM, 80 Ft*lbf)

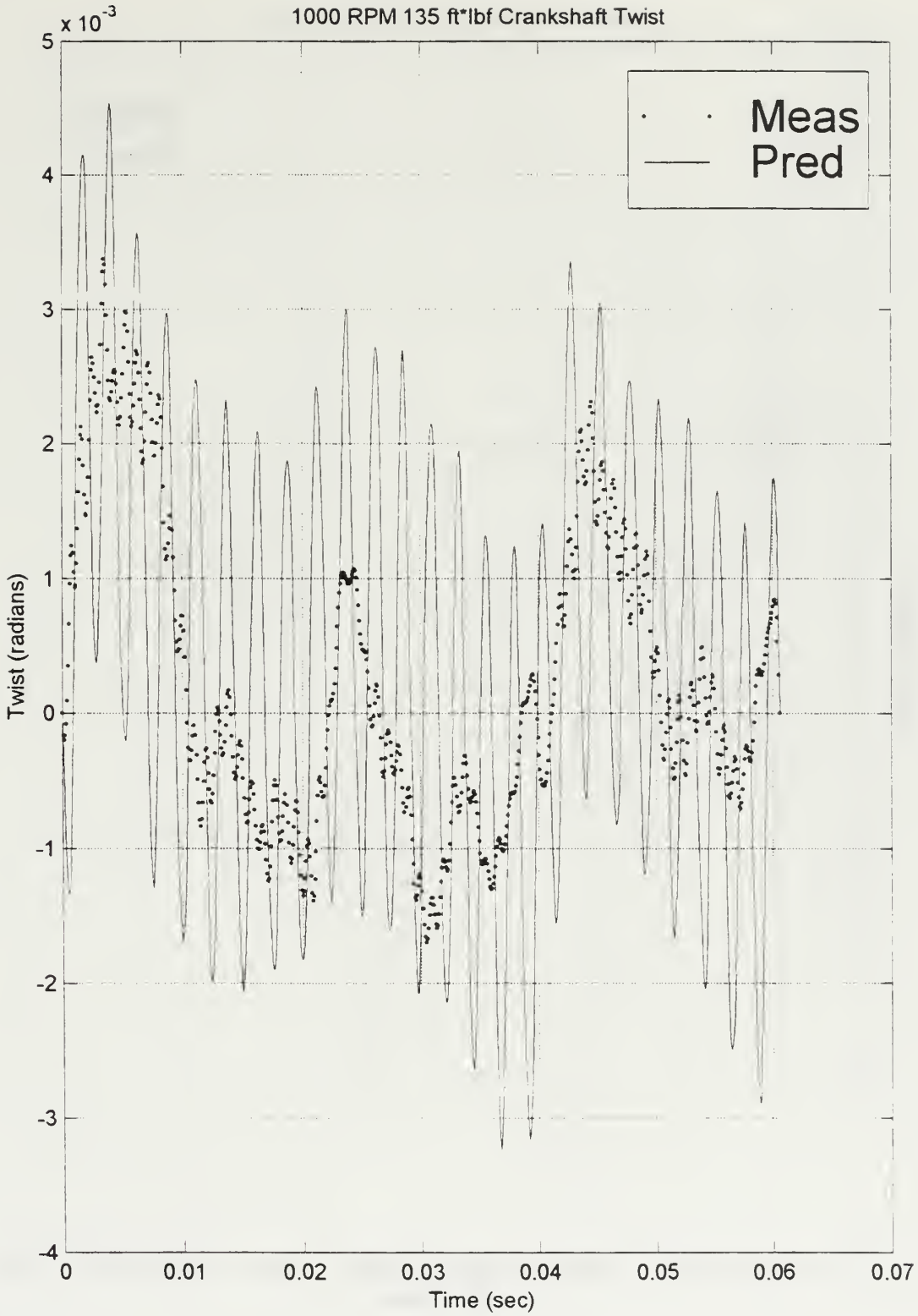


Figure 33. Crankshaft Twist (1000 RPM, 135 Ft*lbf)

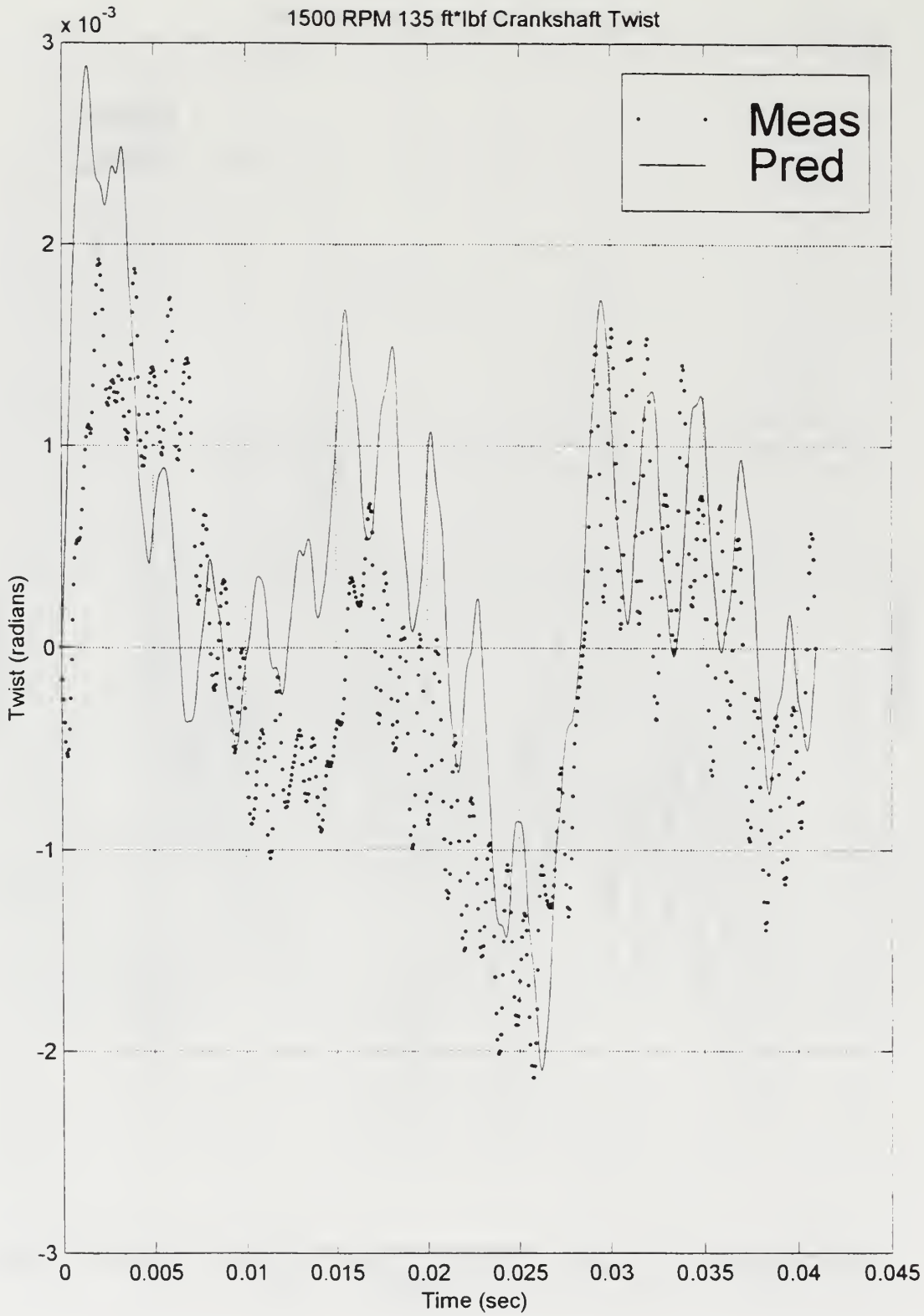


Figure 34. Crankshaft Twist (1500 RPM, 135 Ft*lbf)

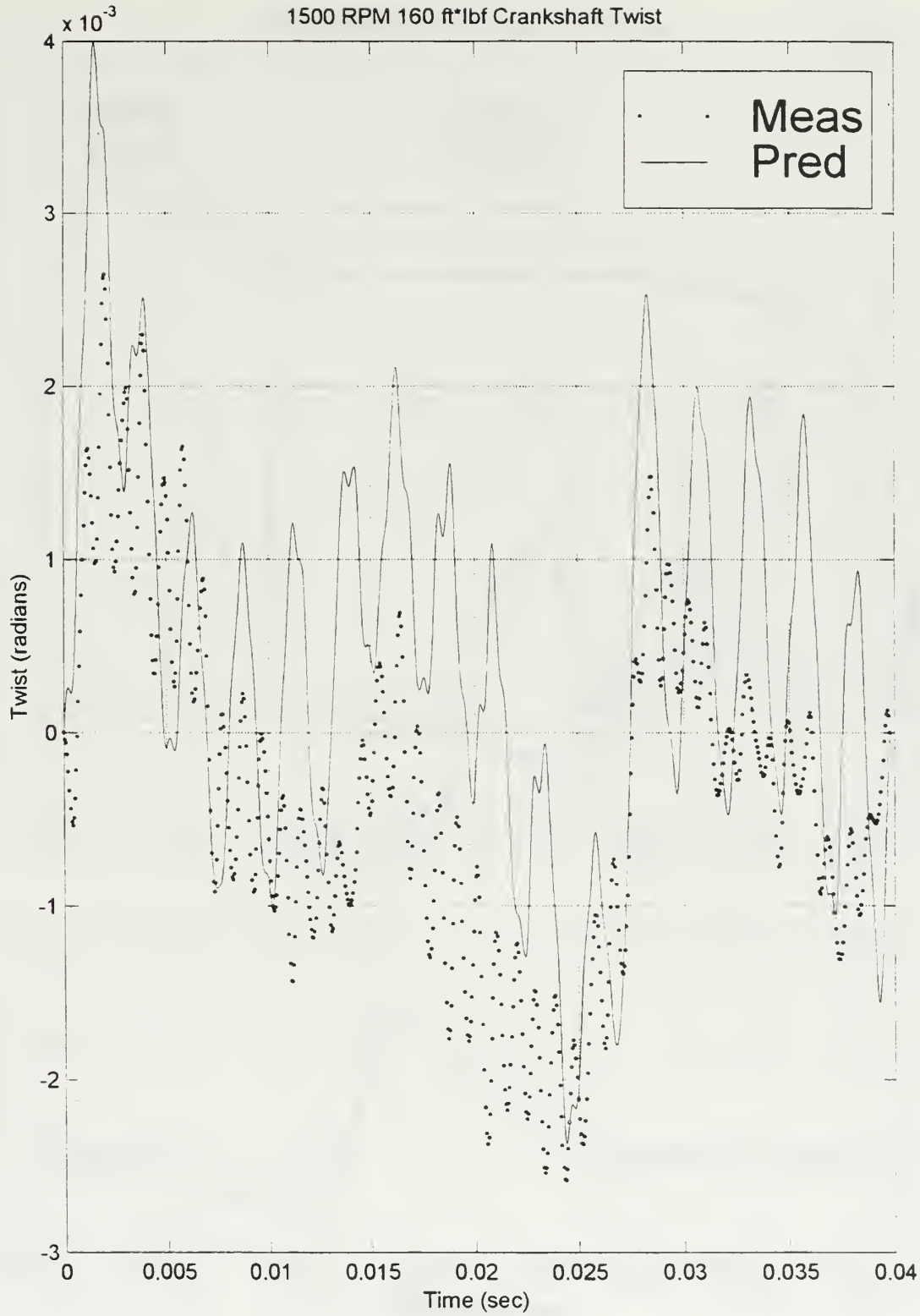


Figure 35. Crankshaft Twist (1500 RPM, 160 Ft*lbf)

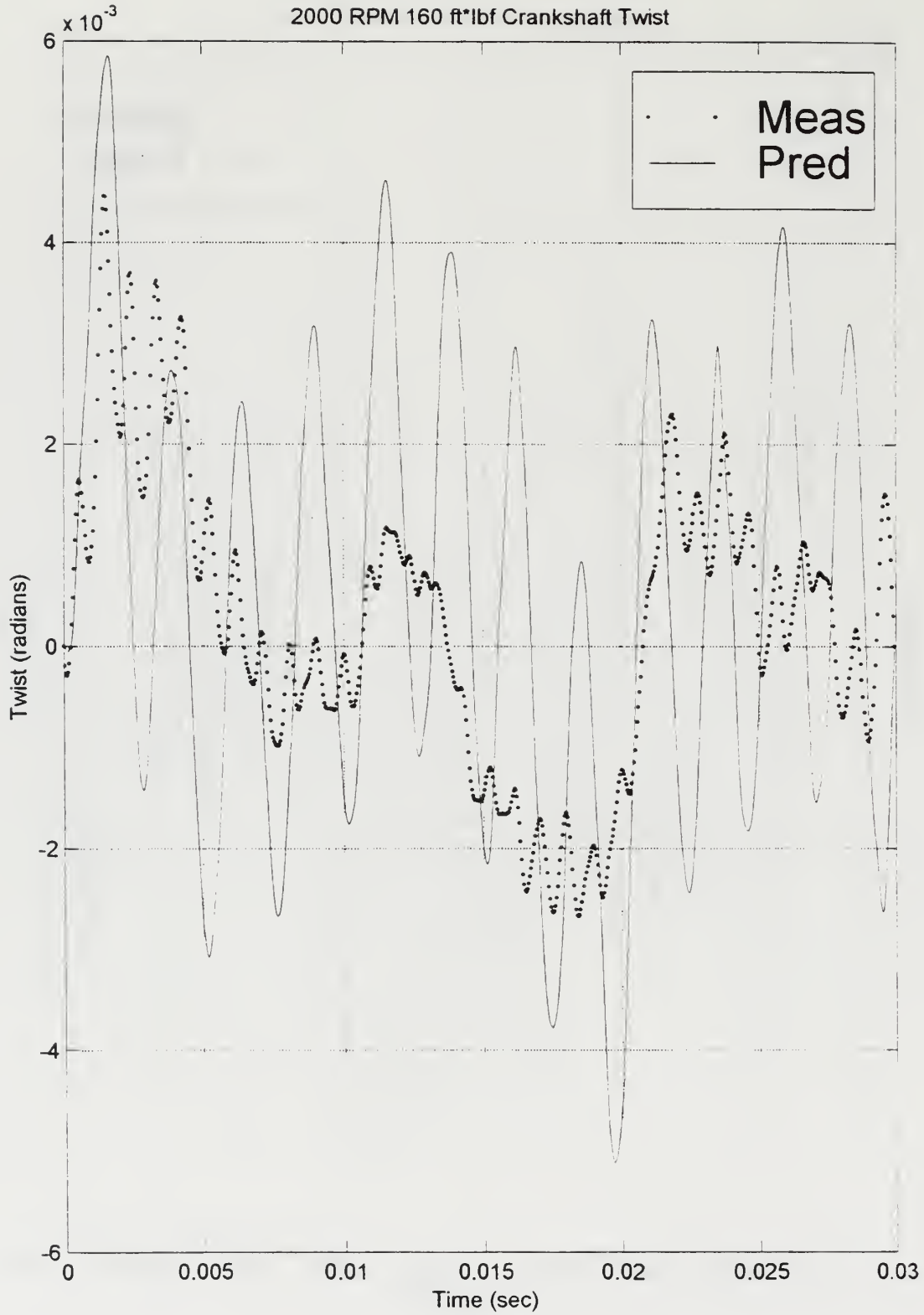


Figure 36. Crankshaft Twist (2000 RPM, 160 Ft*lbf)

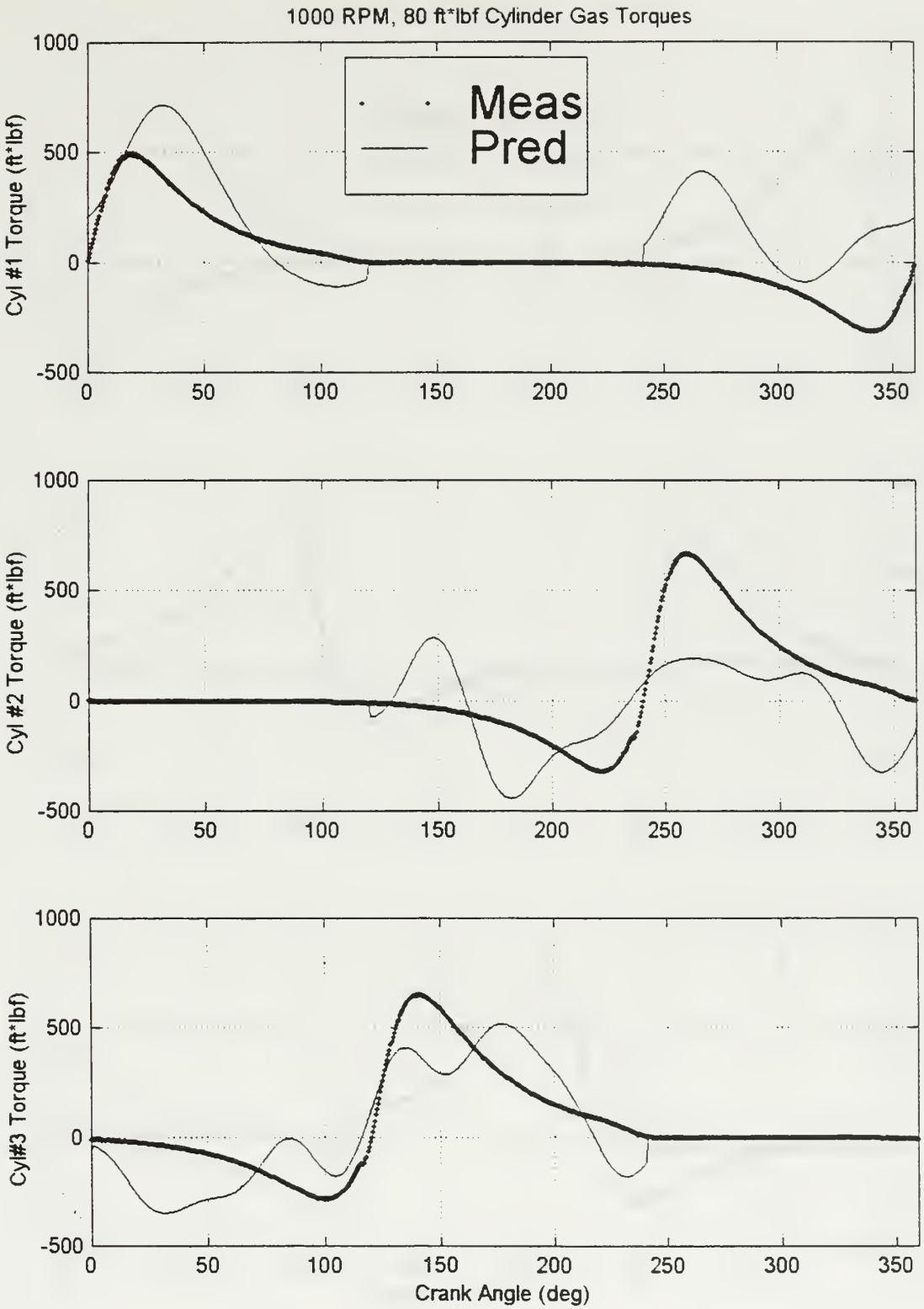


Figure 37. Individual Cylinder Gas Torques (1000 RPM, 80 Ft*lbf)

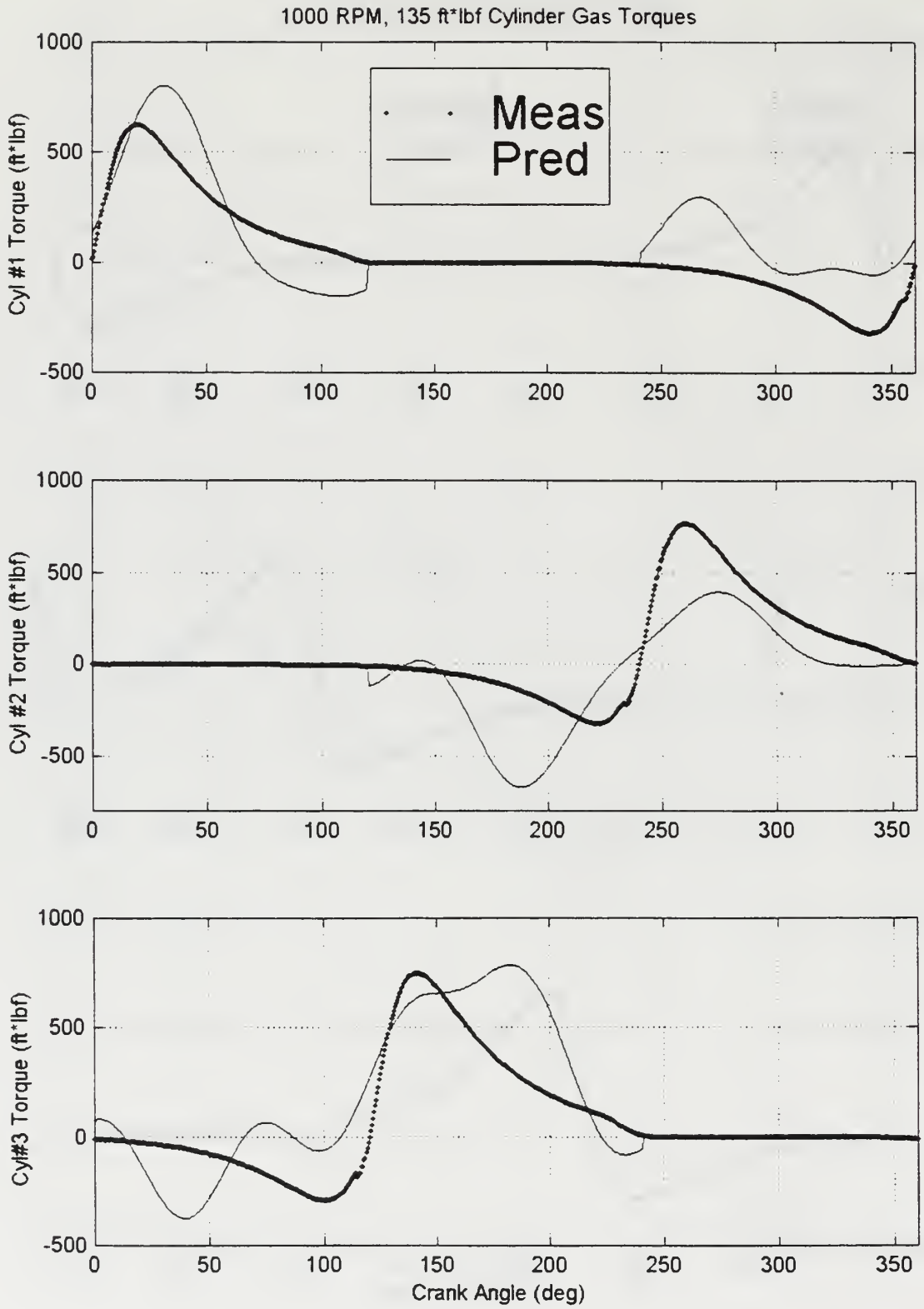


Figure 38. Individual Cylinder Gas Torques (1000 RPM, 135 Ft*lbf)

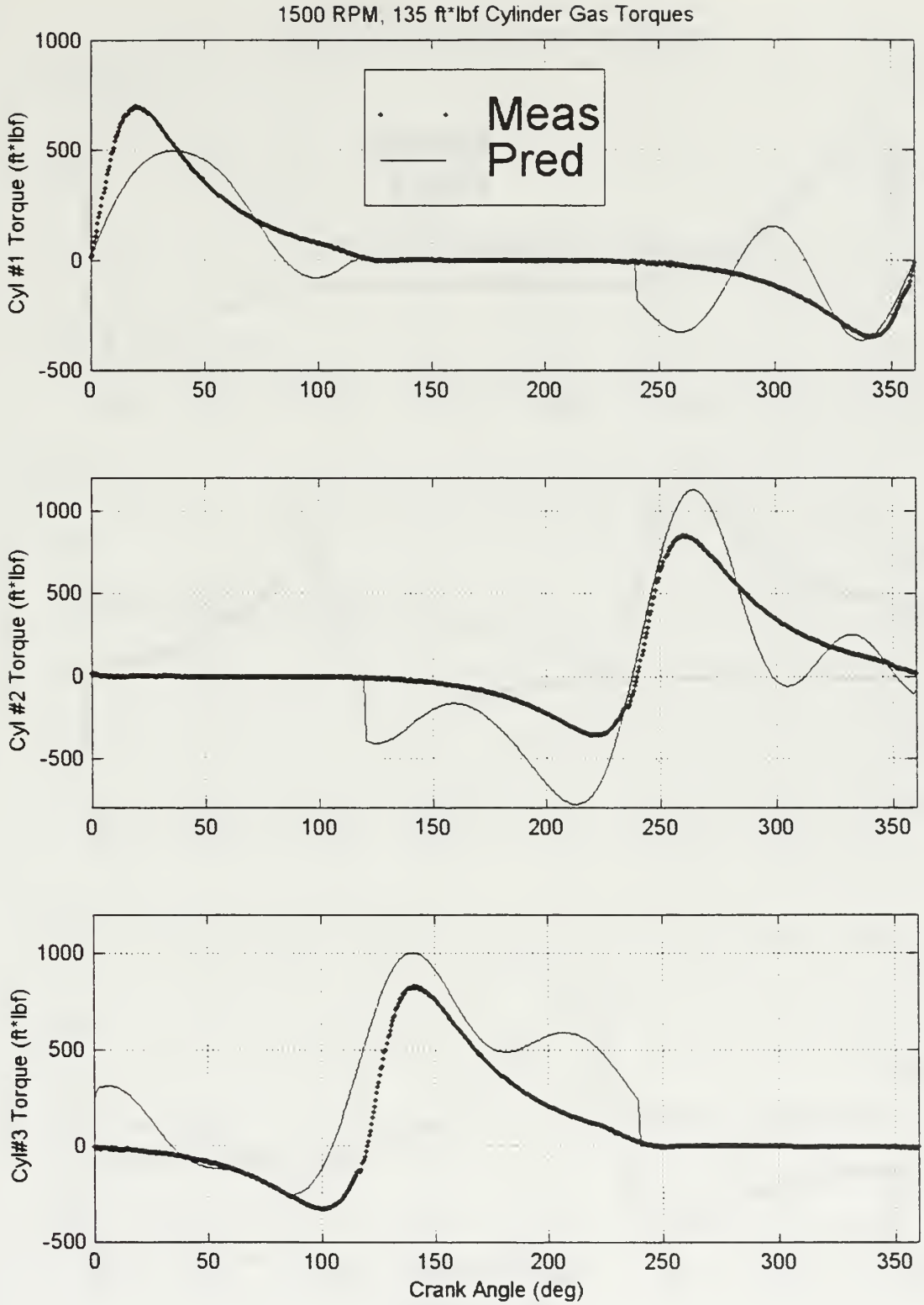


Figure 39. Individual Cylinder Gas Torques (1500 RPM, 135 Ft*lbf)

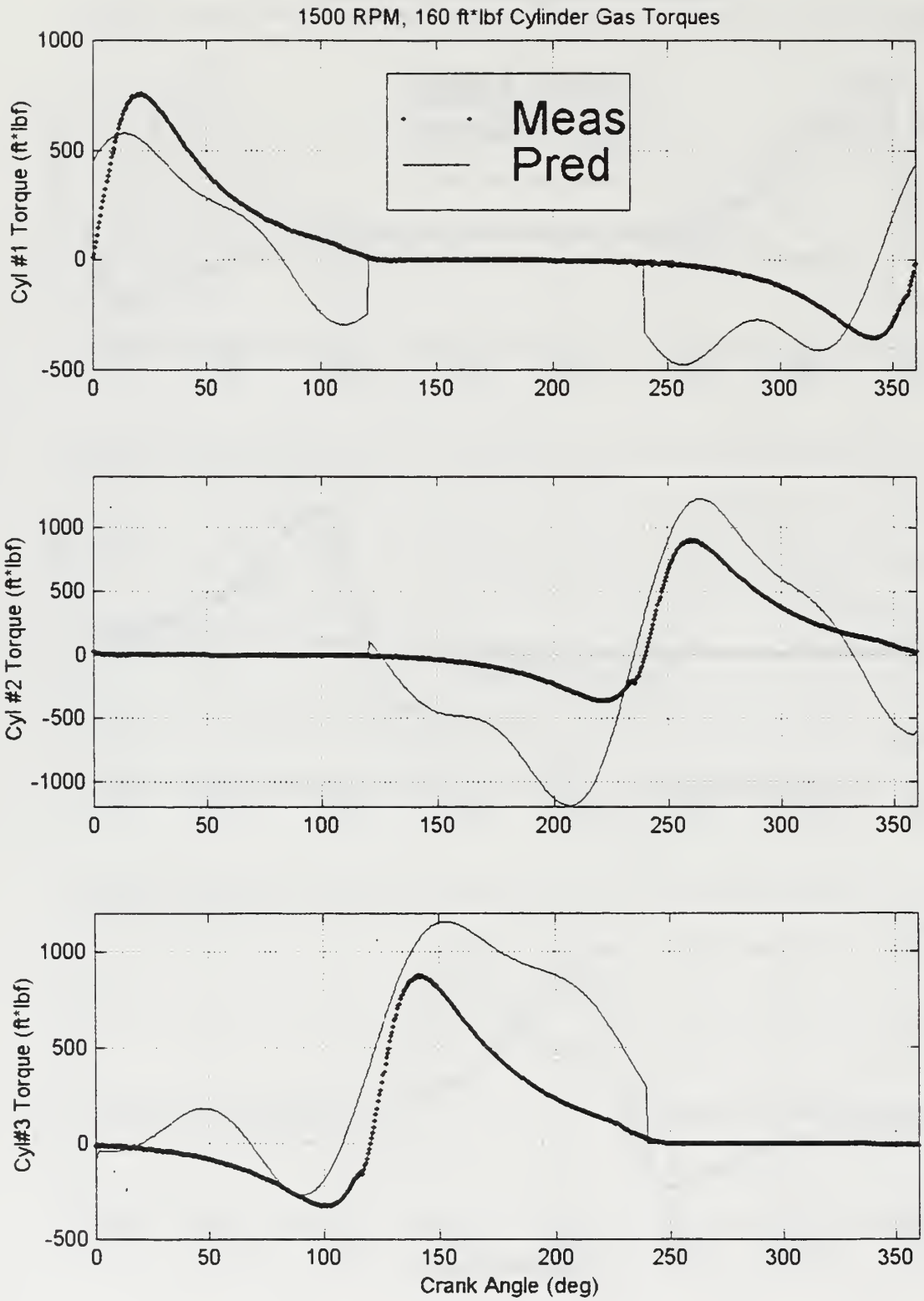


Figure 40. Individual Cylinder Gas Torques (1500 RPM, 160 Ft*lbf)

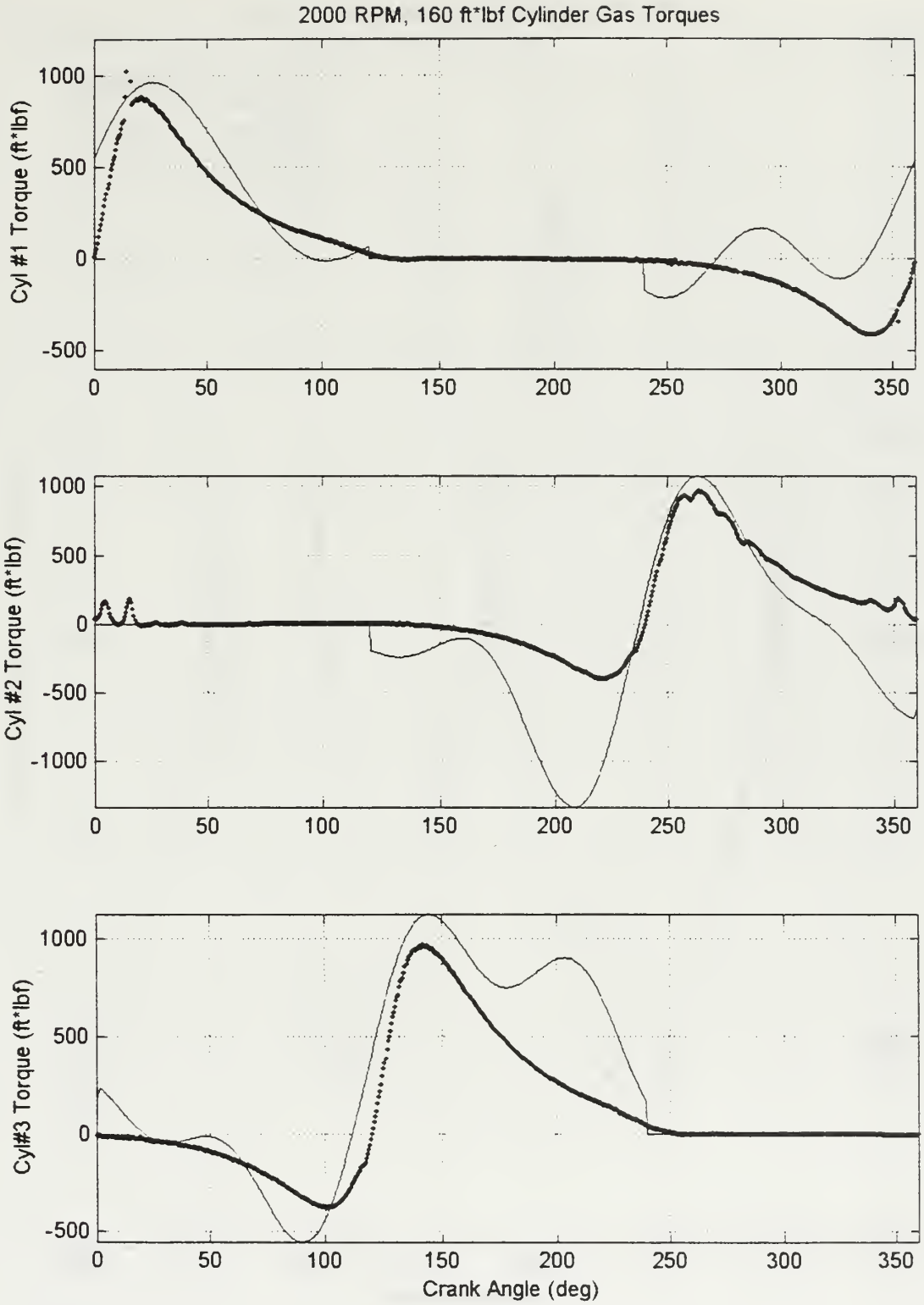


Figure 41. Individual Cylinder Gas Torques (2000 RPM, 160 Ft*lbf)

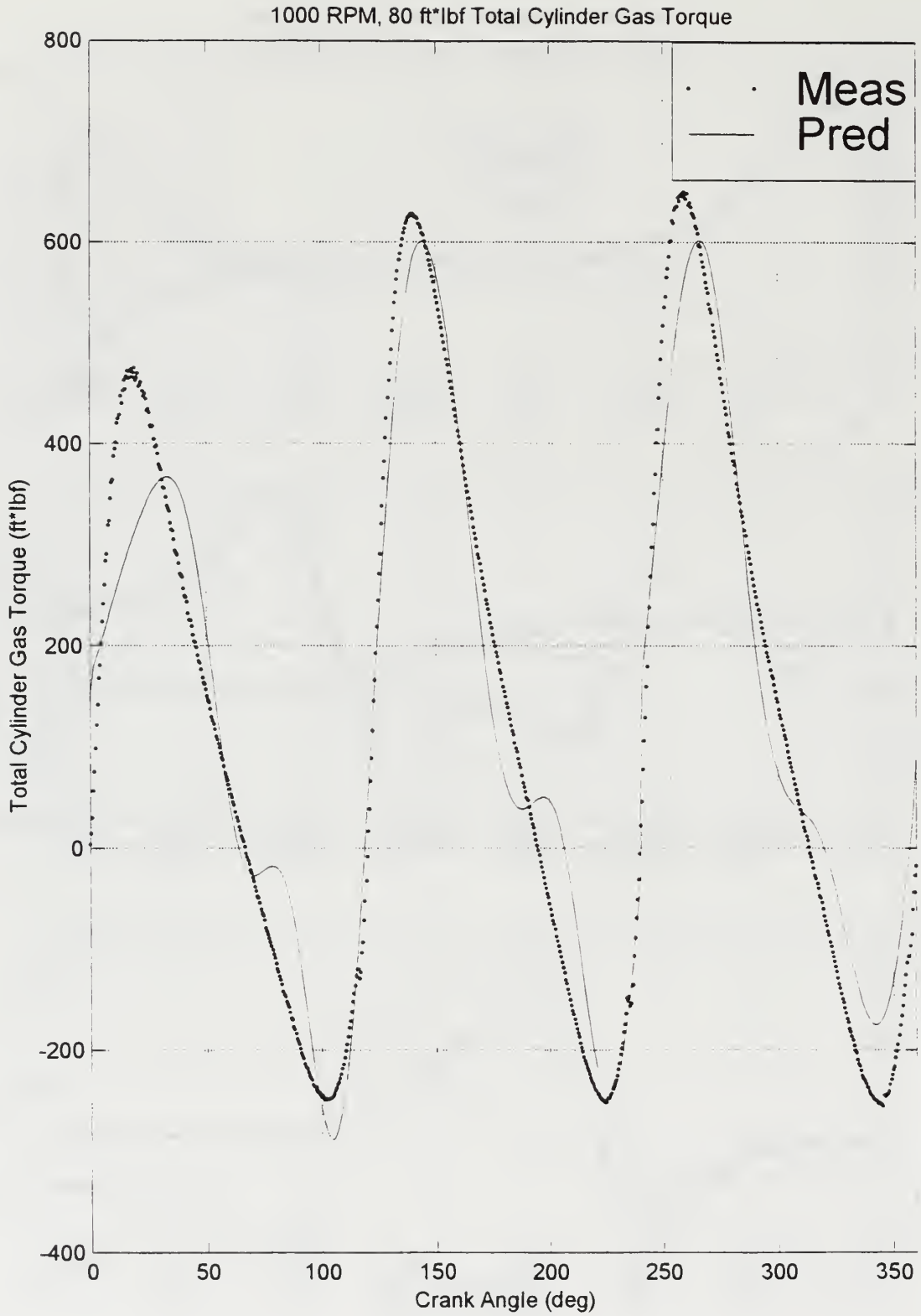


Figure 42. Total Gas Torque (1000 RPM, 80 Ft*lbf)

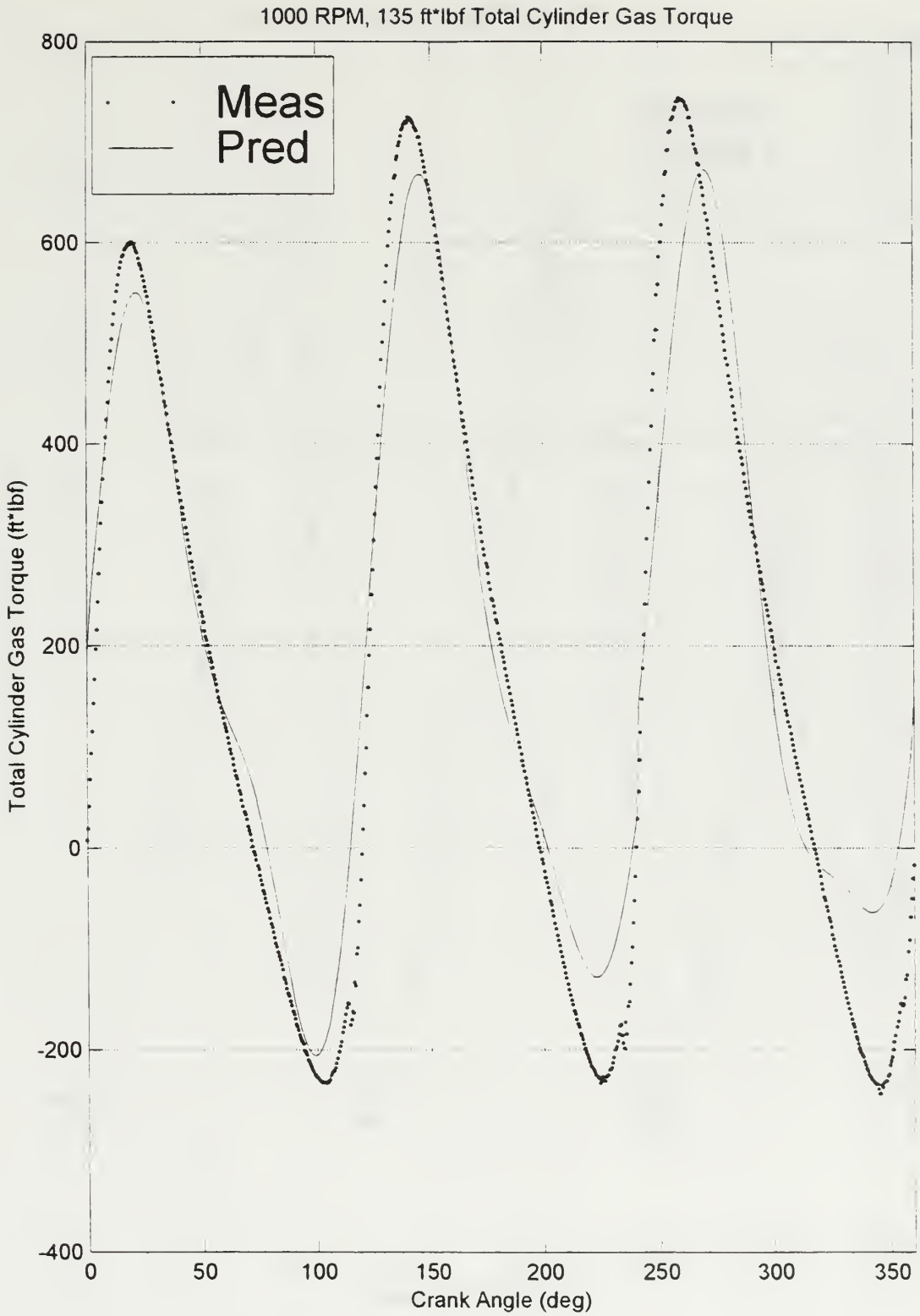


Figure 43. Total Gas Torque (1000 RPM, 135 Ft*lbf)

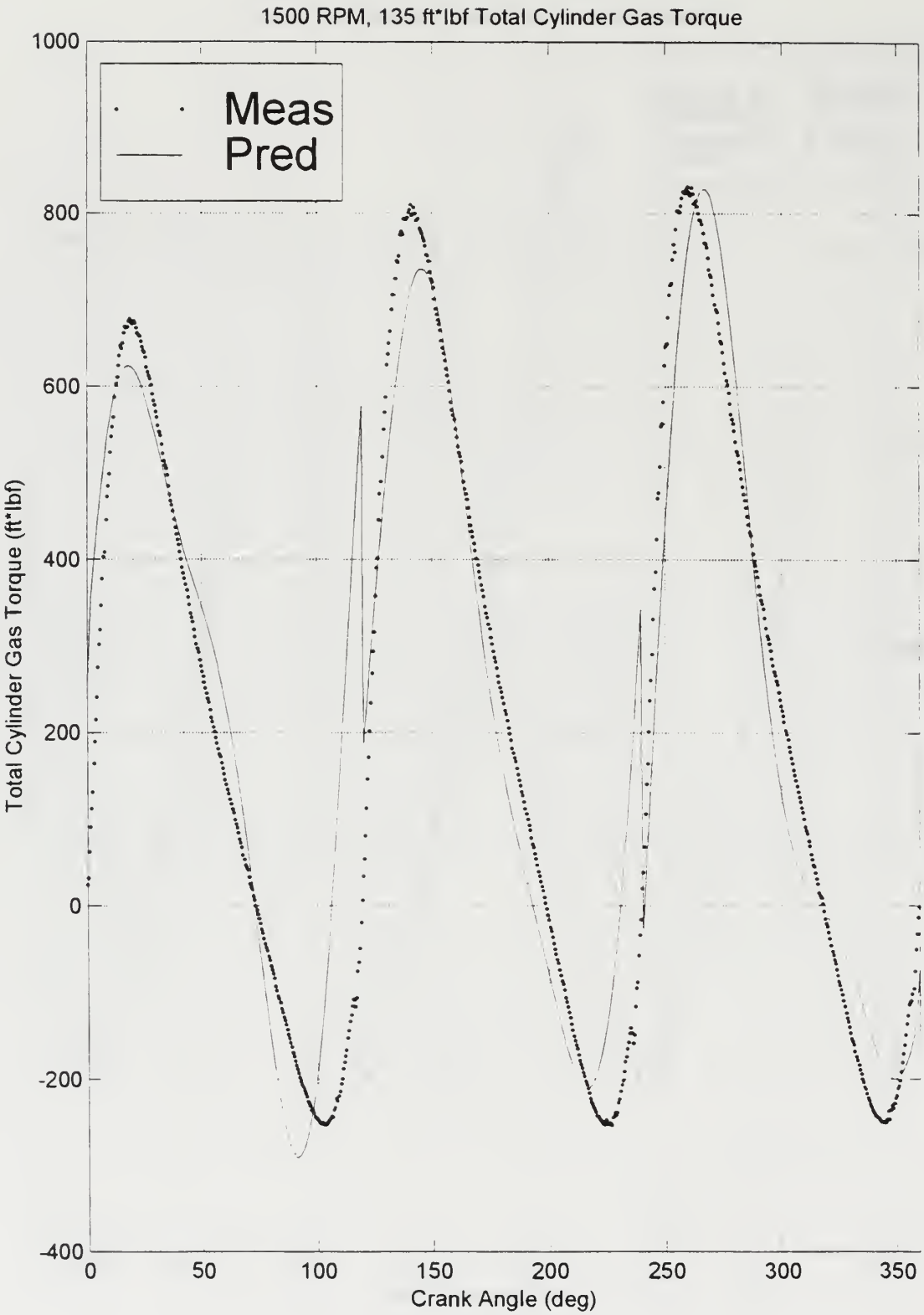


Figure 44. Total Gas Torque (1500 RPM, 135 Ft*lbf)

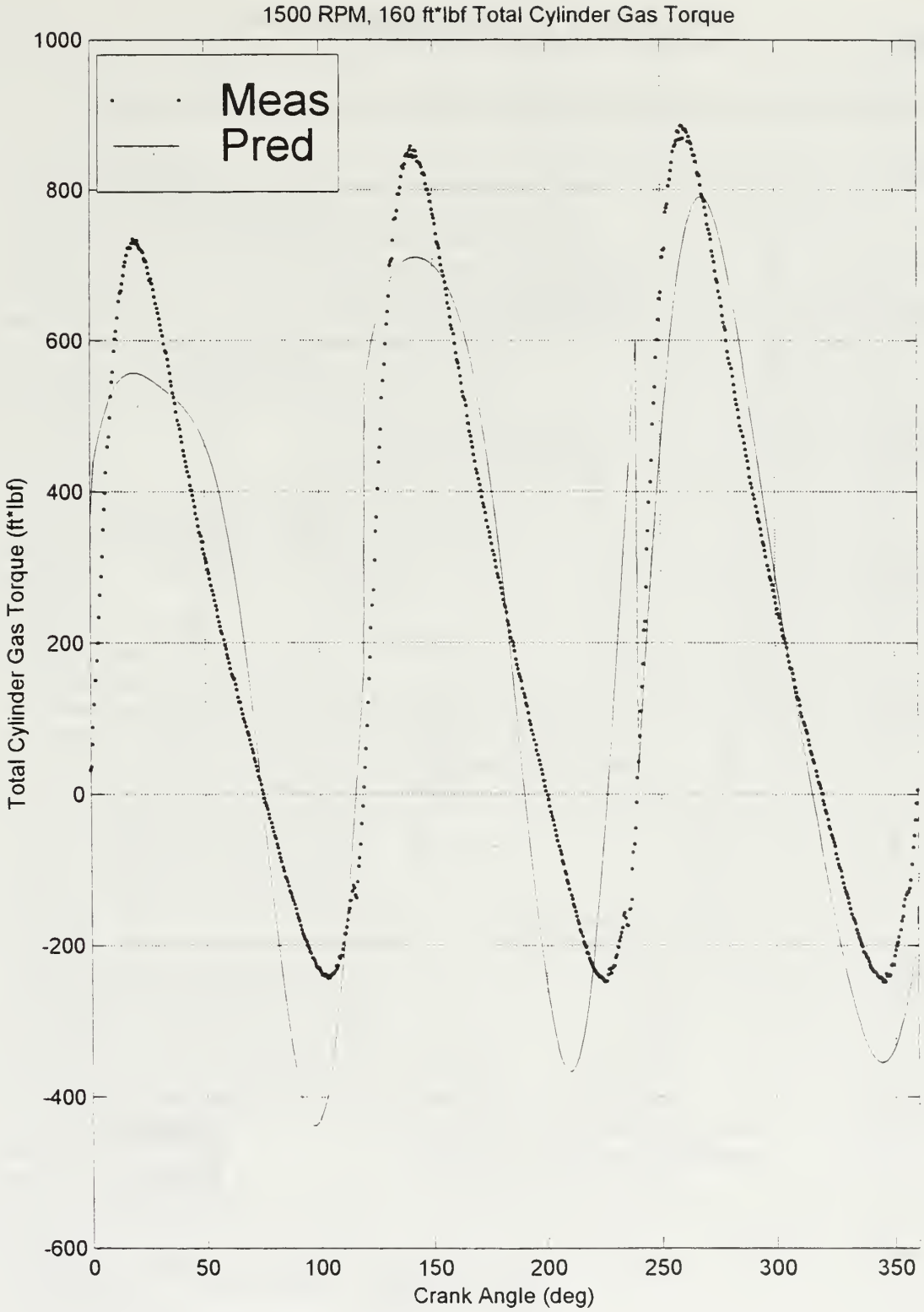


Figure 45. Total Gas Torque (1500 RPM, 160 Ft*lbf)

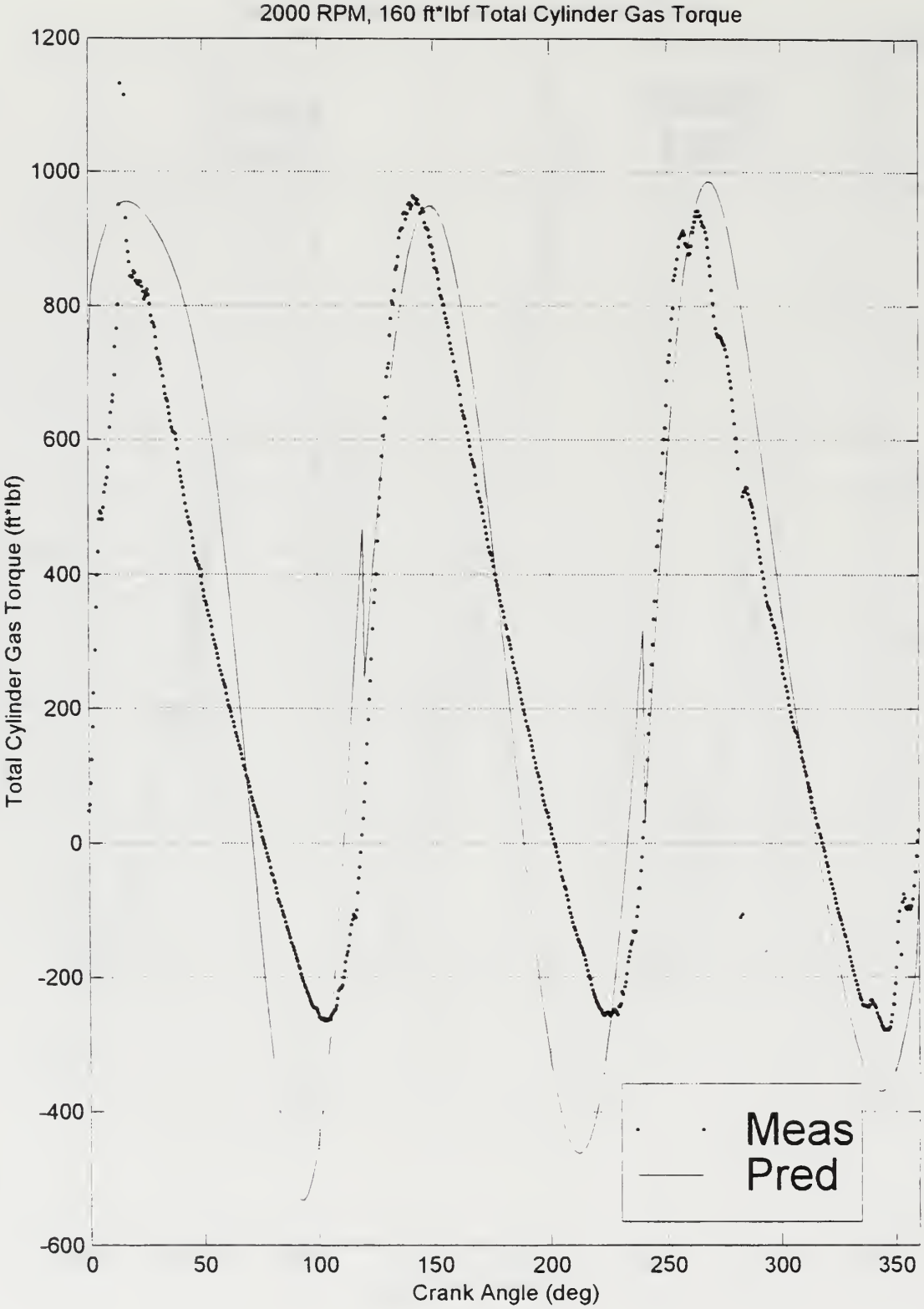


Figure 46. Total Gas Torque (2000 RPM, 160 Ft*lbf)

APPENDIX E. MATLAB CODES

Program used to calculate cylinder gas torques from measured pressures.

```
%       TORQUE36
% This code computes the individual torque contribution of each
% individual cylinder referenced to TDC of Nr.1 Cylinder.
% Gas torque is calculated based on measured pressures
load wa1k100.md                 % ECA cylinder #1 pressure data
load wb1k100.md                 % ECA cylinder #2 pressure data
load wc1k100.md                 % ECA cylinder #3 pressure data
pa = reshape (wa1k100,5,720); P1cyl = [pa(1,:) pa(1)];
pb = reshape (wb1k100,5,720); P2cyl = [pb(1,:) pb(1)];
pc = reshape (wc1k100,5,720); P3cyl = [pc(1,:) pc(1)];
W = 7.556;                      % Reciprocating weight (lbf)
R = 2.25;                       % Crankshaft Eccentricity (in)
B = 3.875;                      % Cylinder Bore (in)
L = 8.8; % Connecting Rod length (in)
g = 386; % Gravitational acceleration (lbf*in/sec^2)
theta = linspace(0,2*pi,721);   % Crank angle vector
N = 1022;                       % RPM
Load = 100;                     % Ft*lbf
omega = 2*pi*N/60;              % rad/sec
dt = (60/N)/720;
s1 = R*cos(theta) + sqrt(L^2 - (R^2)*sin(theta).^2);
s2 = R*cos(theta-4*pi/3) + sqrt(L^2 - (R^2)*sin(theta-4*pi/3).^2);
s3 = R*cos(theta-2*pi/3) + sqrt(L^2 - (R^2)*sin(theta-2*pi/3).^2);
Sp1 = deriv(s1,dt);  Spd1 = deriv(Sp1,dt);% Piston speed (in/sec) and
Sp2 = deriv(s2,dt);  Spd2 = deriv(Sp2,dt);% Piston acceleration (in/sec^2)
Sp3 = deriv(s3,dt);  Spd3 = deriv(Sp3,dt);
T1rec = -(W/g)*Spd1.*(Sp1/omega)/12;
T2rec = -(W/g)*Spd2.*(Sp2/omega)/12;
T3rec = -(W/g)*Spd3.*(Sp3/omega)/12;
Fpar = 201;                     % lbf           Parasitic force
Tpar1 = Fpar*abs(Sp1/omega)/12; % ft*lbf        Parasitic torque
Tpar2 = Fpar*abs(Sp2/omega)/12;
Tpar3 = Fpar*abs(Sp3/omega)/12;
pref = 14.706 + 0.00205*(N-634);  % psia
T1cyl=-((P1cyl.*pref-pref)*(pi*B^2/4).*Sp1/omega)/12;      % FT*LBF
T2cyl=-((P2cyl.*pref-pref)*(pi*B^2/4).*Sp2/omega)/12;      % FT*LBF
T3cyl=-((P3cyl.*pref-pref)*(pi*B^2/4).*Sp3/omega)/12;      % FT*LBF
Tbrg = (0.0003403)*N;                   % FT-LBF Bearing friction per cylinder
Taux = 38;                              % FT-LBF Valve train and auxilliaries
Tpmp = 0.792;                           % FT-LBF Oil pump load
Ttot = T1cyl+T2cyl+T3cyl-Tpar1-Tpar2-Tpar3-3*Tbrg-Taux-Tpmp+T1rec+T2rec+T3rec;% FT-LBF
figure (1)
degrees = theta*180/pi;
plot (degrees,P1cyl,'kx',degrees,P2cyl,'ko',degrees,P3cyl,'k.')
axis([0,360,0,90])
title ('Cylinder Pressures referenced to TDC of #1 Cylinder')
xlabel ('Crank Angle (degrees)')
ylabel ('Cylinder Indicated Pressure (bars, absolute)')
legend ('Cyl#1','Cyl#2','Cyl#3')
```

```
grid
orient landscape
figure(2)
plot(theta,T1cyl,'bX',theta,T2cyl,'bO',theta,T3cyl,'b.')
grid, hold on
plot(theta,T1rec,'rX',theta,T2rec,'rO',theta,T3rec,'r.')
plot (theta,Ttot,'g',theta,Load*ones(size(theta)),'c')
title('Individual Cylinder Torque input Referenced to TDC of Nr. 1 Cylinder')
ylabel('Gas Torque (FT-LB)')
xlabel('Degrees after TDC of NR. 1 Cylinder')
legend('Cyl #1','Cyl #2','Cyl #3')
orient landscape
% COMPUTE AVERAGE TORQUE CONTRIBUTION OF EACH CYLINDER
T1cyl_avg=trapz(theta,T1cyl)/(2*pi)
T2cyl_avg=trapz(theta,T2cyl)/(2*pi)
T3cyl_avg=trapz(theta,T3cyl)/(2*pi)
Avg_Torque_input=trapz(theta,Ttot)/(2*pi)
Torque_out_to_Torque_in = Load/Avg_Torque_input
```

Programs used to calculate angular positions, given cylinder gas torques (time-marching method).

```
%       MEASPRD3
%       Concurrently plots measured and predicted responses
%       using time-marching direct integration method
%
%       Section One plots the measured response
%
load w11k100.md                         % Load optical encoder data file
t = w11k100(:,1);
tt = diff(t);                           % determine dt's
tt = (reshape(tt,720,11))';             % Phase lock one cycle
tttt = mean(tt,1);                      % Ensemble average the phases
position = linspace(0,(2*pi),721);      % The known positions of the O.E. windows
pos = position(1:720);
omega = (1/sum(tttt))*2*pi;             % Mean rotational velocity (rad/sec)
timem = [0 cumsum(tttt)];               % Time vector corresponding to position
angposm = position-timem.*omega;        % Angular position (radians)
plot(timem,angposm,'rO')                % Plot measured angular position vs time
hold on
%
%       Section Two plots the predicted response based on pressure data
%       Uses a SIX SECOND ORDER SUMULTANEOUS EQUATION ODE SOLVER
%       Calls "deqns.m" which defines the system of 2nd order ode's
%
global Tload T1cyl T2cyl T3cyl Taux Tpmp j2rec j3rec j4rec i
load wa1k100.md         % ECA cylinder #1 pressure data
load wb1k100.md         % ECA cylinder #2 pressure data
load wc1k100.md         % ECA cylinder #3 pressure data
pa = reshape (wa1k100,5,720); pa = pa(1,:);
pb = reshape (wb1k100,5,720); pb = pb(1,:);
pc = reshape (wc1k100,5,720); pc = pc(1,:);
P1cyl = [pa(1,:) pa(1)];% Cylinder pressures (bars)
```

```
P2cyl = [pb(1,:) pb(1)];
P3cyl = [pc(1,:) pc(1)];
shp = [0 0 0 0 0 -1.8]*1e-03;
shv = [1 1 1 1 1 1]*106.3;
ic = [shp shv];                    % define IC's
Tload=1200;                        % lbf*in Load torque
timep=linspace(0,sum(tttt),721);   % divide one rev into 720 divisions
angposp = zeros(1,721);            % initialize predicted position vector
B = 3.875;                         % in     Cylinder bore
R = 2.25;                          % in     Crankshaft eccentricity
W = 7.556;                         % lbf    Weight of reciprocating components
g = 386;                           % in/sec^2       gravitational acceleration
L = 8.8;                           % in     Connecting rod length
N = omega*60/(2*pi);               % RPM
pos = linspace (0,(2*pi),721);
pref = 14.706 + 0.00205*(N-634);   % psia
dt = (2*pi/omega)/720;
s1 = R*cos(pos) + sqrt(L^2 - (R^2)*sin(pos).^2);
s2 = R*cos(pos-4*pi/3) + sqrt(L^2 - (R^2)*sin(pos-4*pi/3).^2);
s3 = R*cos(pos-2*pi/3) + sqrt(L^2 - (R^2)*sin(pos-2*pi/3).^2);
Sp1 = deriv(s1,dt);  Spd1 = deriv(Sp1,dt);    % Piston speed (in/sec) and
Sp2 = deriv(s2,dt);  Spd2 = deriv(Sp2,dt);    % Piston acceleration (in/sec^2)
Sp3 = deriv(s3,dt);  Spd3 = deriv(Sp3,dt);
Fpar = 100;                        %      lbf*in   Parasitic force
Tpar1 = Fpar*abs(Sp1/omega);       %      lbf*in   Parasitic torque
Tpar2 = Fpar*abs(Sp2/omega);
Tpar3 = Fpar*abs(Sp3/omega);
pos = linspace (0,(2*pi),721);
pref = 14.706 + 0.00205*(N-634);   % psia
Tpmp = 9.5;        %lbf*in Oil pump torque
T1cyl = -(P1cyl.*pref-pref)*(pi*B^2/4).*Sp1/omega; % (in*lbf)
T2cyl = -(P2cyl.*pref-pref)*(pi*B^2/4).*Sp2/omega; % (in*lbf)
T3cyl = -(P3cyl.*pref-pref)*(pi*B^2/4).*Sp3/omega; % (in*lbf)
Taux = 168;        %lbf*in Valvetrain and auxilliary torque
j2rec = (W*R^2/(2*g))*(1-cos(2*pos));              %lbf*in*sec^2
j3rec = (W*R^2/(2*g))*(1-cos(2*(pos-4*pi/3)));     %lbf*in*sec^2
j4rec = (W*R^2/(2*g))*(1-cos(2*(pos-2*pi/3)));     %lbf*in*sec^2
T1rec = -(W/g)*Spd1.*(Sp1/omega);
T2rec = -(W/g)*Spd2.*(Sp2/omega);
T3rec = -(W/g)*Spd3.*(Sp3/omega);
T1cyl = T1cyl - Tpar1 + T1rec;
T2cyl = T2cyl - Tpar2 + T2rec;
T3cyl = T3cyl - Tpar3 + T3rec;
step = 1;
theta = zeros(721,6);
theta(1,:) = shp;
for i = 1:step:720;
  [T,x] = ode45('deqns',[timep(i) timep(i+step)],ic);
  ic = x(length(T),:);             % reinitialize IC's from previous iteration
  theta(i+step,:) = x(length(T),1:6);
  angposp(i+step) = x(length(T),1) - omega*timep(i+step);
end
plot(timep,angposp,'bX')  % Plot predicted angular position vs time
title('Time Marching ODE45 Method Comparison to Measured Data')
xlabel('Time (sec)')
```

```
ylabel('Theta one Phase Deviation (radians)')
legend ('Meas      ','Pred      ')
grid
orient landscape
shaft_pos = (ic(1:6)-2*pi)'
cycle_stat = (ic - [(shp+2*pi) shv])'
% Plot predicted angular velocity
figure (2)
omegap = [shv(1) (diff(angposp)./timep(2)+omega)];
omegam = (1./(720*tttt))*2*pi;
omegam = [omegam omegam(720)];
plot(timep,omegap,'b',timep,omegam,'r')
hold on
plot(timep,omega*ones(size(timep)))
xlabel('time(sec)')
ylabel('Angular Velocity (rad/sec)')
grid, orient tall
figure (3)
plot(timem,angposm,'k.')
hold on
plot(timep,angposp,'k')
title('Time Marching ODE45 Method Comparison to Measured Data')
xlabel('Time (sec)')
ylabel('Theta one Phase Deviation (radians)')
legend ('Meas      ','Pred      ')
grid

% DEQNS function to determine six second order differential equations
% To be used in ode45 fctn in measpred program
%----------------------------
function xdot=deqns(t,x)
%----------------------------
global Tload T1cyl T2cyl T3cyl Taux Tpmp j2rec j3rec j4rec i
%------Constants to be used in differential equations------
j1=.02443;        %lb*in*sec^2/rad
j2=.2482;         %lb*in*sec^2/rad
j3=.1462;         %lb*in*sec^2/rad
j4=.2482;         %lb*in*sec^2/rad
j5=7.222;         %lb*in*sec^2/rad
j6=.2870;         %lb*in*sec^2/rad
k1=3.11e6;        %lb*in/rad
k2=7.00e6;        %lb*in/rad
k3=7.00e6;        %lb*in/rad
k4=10.82e6;       %lb*in/rad
k5=1.304e6;       %lb*in/rad
c12=.01;          %lb*in*sec/rad
c23=.01;          %lb*in*sec/rad
c34=.01;          %lb*in*sec/rad
c45=.01;          %lb*in*sec/rad
c56=.01;          %lb*in*sec/rad
c2=0.013;         %lb*in*sec/rad
c3=0.013;         %lb*in*sec/rad
c4=0.013;         %lb*in*sec/rad
%-------------------------------------------------------------
%         12 first order equations which define the
```

```
%       original 6 second order equations of motion
%-------------------------------------------------------------
xdo(1)=x(7);
xdo(2)=x(8);
xdo(3)=x(9);
xdo(4)=x(10);
xdo(5)=x(11);
xdo(6)=x(12);
xdo(7)=((c12*x(8))-(c12*x(7))+(k1*x(2))-(k1*x(1))-Tpmp)/j1;
xdo(8)=(T1cyl(i)+(c23*x(9))-((c23+c12+c2)*x(8))+(c12*x(7))+(k2*x(3))...
  -((k1+k2)*x(2))+(k1*x(1)))/(j2+j2rec(i));
xdo(9)=(T2cyl(i)+(c34*x(10))-((c34+c23+c3)*x(9))+(c23*x(8))+(k3*x(4))...
  -((k2+k3)*x(3))+(k2*x(2)))/(j3+j3rec(i));
xdo(10)=(T3cyl(i)+(c45*x(11))-((c45+c34+c4)*x(10))+(c34*x(9))+(k4*x(5))...
  -((k3+k4)*x(4))+(k3*x(3)))/(j4+j4rec(i));
xdo(11)=((c56*x(12))+((c56+c45)*x(11))+(c45*x(10))+(k5*x(6))...
  -((k4+k5)*x(5))+(k4*x(4))-Taux)/j5;
xdo(12)=(-Tload-(c56*x(12))+(c56*x(11))-(k5*x(6))+(k5*x(5)))/j6;
xdot=xdo'; % vector defining the equations of motion
```

Programs used to calculate angular positions, given measured cylinder gas torques (finite element method).

```
%       MEASPS
%       Determines and plots comparisons of the following:
%       (1) Measured response from flywheel and optical encoder data
%       (2) Predicted response from inertial model based on measured
%                       pressure data from the three cylinders
% Evaluates predicted response using a finite element formulation
%-------------------------------------------
%       Section One plots the measured response
%-------------------------------------------
%-----Measured response from flywheel-----
load w51k100.md                         % Load time data from MDA
t = [0;w51k100(1:7938,1)];              % Extract time data only
tt = diff(t);                           % COMPUTES THE DEL_T'S
tt = (reshape(tt,126,63))';             % Phase lock one cycle
tttt = mean(tt);                        % Ensemble average the phases
dtrat = tt(1)/mean(tt(2:63,1));
teeth = linspace(0,2*pi,127); tooth = 2*pi/127;
th51 = tooth*dtrat;
pos5 = [0 teeth(1:125)+th51];
pos5(127) = 2*pi;
time5 = [0,cumsum(tttt)-(1-dtrat)*tttt(1)];
time5(127) = sum(tttt);
omega = (1/max(time5))*2*pi;
angpos5 = pos5-time5.*omega;%Angular position (radians)
figure (1)
subplot (2,1,1)
plot(time5,angpos5,'k.')                % Plot measured angular position vs time
hold on
%-----Measured response from optical encoder-----
load w11k100.md                         % Load time data from MDA
t = [0;w11k100(1:7920,1)];              % Extract time data only
```

```
tt = diff(t);                                % COMPUTES THE DEL_T'S
tt = (reshape(tt,720,11))';                  % Phase lock one cycle
tttt = mean(tt);                             % Ensemble average the phases
pos1 = linspace(0,(2*pi),721);               % Crank angle position for O.E. windows
time1 = [0 cumsum(tttt)];                    % Time vector corresponding to position
time1 = time1*(max(time5)/max(time1));       % Adjust times to agree
angpos1 = pos1-time1.*omega;                 %Angular position (radians)
figure (1)
subplot (2,1,2)
plot(time1,angpos1,'k.')                     % Plot measured angular position vs time
hold on
%-----------------------------------------------------------------
%       Section Two plots the predicted response based on pressure data
%-----------------------------------------------------------------
global j2rec j3rec j4rec
%------ Cylinder pressures (bars) ------
load wa1k100.md          % ECA cylinder #1 pressure data
load wb1k100.md          % ECA cylinder #2 pressure data
load wc1k100.md          % ECA cylinder #3 pressure data
pa = reshape (wa1k100,5,720); P1cyl = [pa(1,:) pa(1)];
pb = reshape (wb1k100,5,720); P2cyl = [pb(1,:) pb(1)];
pc = reshape (wc1k100,5,720); P3cyl = [pc(1,:) pc(1)];
% Variable descriptions
%  k = element matrix
%  f = element vector
%  kk = compressed system matrix
%  ff = system vector
%  bcdof = a vector containing dofs associated with boundary conditions
%  bcval = a vector containing boundary condition values associated with
%       the dofs in 'bcdof'
%-------------------------------------
%  input data for control parameters
%-------------------------------------
nel = 720;                          % number of elements
nnel = 2;                           % number of nodes per element
ndof = 6;                           % number of dofs per node
nnode = 721;                        % total number of nodes in system
sdof = nnode*ndof;                  .% total system dofs
Tload = 1200;                       % (lbf*in)
R = 2.25;                           % (in)          Crankshaft eccentricity
W = 7.556;                          % (lbf)         Weight of reciprocating components
g = 386;                            % (in/sec^2)    Gravitational acceleration
B = 3.875;                          % (in)          Cylinder Bore
L = 8.8;                            % (in)          Connecting Rod length
N = omega*60/(2*pi);                % RPM
dt = (2*pi/omega)/720;
s1 = R*cos(pos1) + sqrt(L^2 - (R^2)*sin(pos1).^2);
s2 = R*cos(pos1-4*pi/3) + sqrt(L^2 - (R^2)*sin(pos1-4*pi/3).^2);
s3 = R*cos(pos1-2*pi/3) + sqrt(L^2 - (R^2)*sin(pos1-2*pi/3).^2);
Sp1 = deriv(s1,dt);  Spd1 = deriv(Sp1,dt);   % Piston speed (in/sec) and
Sp2 = deriv(s2,dt);  Spd2 = deriv(Sp2,dt);   % Piston acceleration (in/sec^2)
Sp3 = deriv(s3,dt);  Spd3 = deriv(Sp3,dt);
Fpar = 100;                         %       lbf*in    Parasitic force
Tpar1 = Fpar*abs(Sp1/omega);        %       lbf*in    Parasitic torque
Tpar2 = Fpar*abs(Sp2/omega);
```

```
Tpar3 = Fpar*abs(Sp3/omega);
pos = linspace (0,(2*pi),721);
pref = 14.706 + 0.00205*(N-634);  % psia
Tpmp = 9.5;       %lbf*in Oil pump torque
T1cyl = -(P1cyl.*pref-pref)*(pi*B^2/4).*Sp1/omega;          % (in*lbf)
T2cyl = -(P2cyl.*pref-pref)*(pi*B^2/4).*Sp2/omega;          % (in*lbf)
T3cyl = -(P3cyl.*pref-pref)*(pi*B^2/4).*Sp3/omega;          % (in*lbf)
Taux = 168;       %lbf*in Valvetrain and auxilliary torque
j2rec = (W*R^2/(2*g))*(1-cos(2*pos));                       %lbf*in*sec^2
j3rec = (W*R^2/(2*g))*(1-cos(2*(pos-4*pi/3)));              %lbf*in*sec^2
j4rec = (W*R^2/(2*g))*(1-cos(2*(pos-2*pi/3)));              %lbf*in*sec^2
T1rec = -(W/g)*Spd1.*(Sp1/omega);
T2rec = -(W/g)*Spd2.*(Sp2/omega);
T3rec = -(W/g)*Spd3.*(Sp3/omega);
T1cyl = T1cyl - Tpar1 + T1rec;
T2cyl = T2cyl - Tpar2 + T2rec;
T3cyl = T3cyl - Tpar3 + T3rec;
pack
%-----------------------------------------
%  input data for nodal coordinate values
%-----------------------------------------
tcoord = linspace(0,max(time5),721);
 %-------------------------------------------------------
%  input data for nodal connectivity for each element
%-------------------------------------------------------
nodes = [(1:nel)',(2:nnode)'];
%--------------------------------------
%  input data for boundary conditions
%--------------------------------------
%          Dirichlet Boundary Conditions
shp = [0 0 0 0 0 -200]*1e-05;
bcdof = [1,2,3,4,5,6,sdof-5,sdof-4,sdof-3,sdof-2,sdof-1,sdof];
bcval = [shp (shp+2*pi)];
%----------------------------------------
%  initialization of matrices and vectors
%----------------------------------------
ff = zeros(sdof,1);          % initialization of system force vector
kk = zeros(sdof,18);         % initialization of compressed system matrix
index = zeros(sdof,1);       % initialization of kk index vector
%-------------------------------------------------------------------
%  computation of element matrices and vectors and their assembly
%-------------------------------------------------------------------
for i=1:nel;                              % loop for the total number of elements
  nl=nodes(i,1); nr=nodes(i,2);           % extract nodes for (iel)-th element
  tl=tcoord(nl); tr=tcoord(nr);           % extract nodal coord values for the element
  k = Kelm(tl,tr,i);                              % compute element matrix
  f = ((tr-tl)/2)*[-Tpmp; T1cyl(i); T2cyl(i); T3cyl(i); -Taux; -Tload;...
            -Tpmp; T1cyl(i+1); T2cyl(i+1); T3cyl(i+1); -Taux; -Tload];
                                          % compute element vector
  for ii = 1:12;                          % assemble element matrices and vectors
    ff((i-1)*6+ii) = ff((i-1)*6+ii) + f(ii);
    if ii<=6;
      kk((i-1)*6+ii,:) = kk((i-1)*6+ii,:) + [0,0,0,0,0,0,k(ii,:)];
      index(i*6+ii) = (i-1)*6;
    else
```

```
      kk((i-1)*6+ii,:) = kk((i-1)*6+ii,:) + [k(ii,:),0,0,0,0,0,0];
    end
  end
end
index(1:6) = -6*[1;1;1;1;1;1];
%-----------------------------
%  apply boundary conditions
%-----------------------------
%        Dirichlet Boundary Conditions
for i = 1:length(bcdof);
  kk(bcdof(i),:) = zeros(1,18);
  kk(bcdof(i),bcdof(i)-index(bcdof(i))) = 1;
  ff(bcdof(i)) = bcval(i);
end
%-----------------------------------------
%        Transform kk and ff to upper diagonal
%-----------------------------------------
for i = 1:(sdof-1);
  f = fix((i-1)/6)*6+12;
  if i>(sdof-6); f=sdof; end
  for ii = (i+1):f;                          % other rows with data in column i
    v = kk(ii,i-index(ii))./kk(i,i-index(i));    % multiplier
    kk(ii,i-index(ii)) = 0;
    for j = (i+1):f;                         % data elements in row ii
      kk(ii,j-index(ii)) = kk(ii,j-index(ii))-v*kk(i,j-index(i));
    end
    ff(ii) = ff(ii) - v*ff(i);
  end
end
%-----------------------------
%        Solve matrix eqn for theta
%-----------------------------
theta = zeros(1,sdof);
theta(sdof) = ff(sdof)/kk(sdof,sdof-index(sdof));
for i = (sdof-1):(-1):1;
  f = fix((i-1)/6)*6+12;
  if i>(sdof-6); f=sdof; end
  for j = (i+1):f;
    ff(i) = ff(i) - kk(i,j-index(i))*theta(j);
  end
  theta(i) = ff(i)/kk(i,i-index(i));
end
theta = reshape(theta,6,nnode);
angposp1 = theta(1,:) - omega*tcoord;
angposp5 = theta(5,:) - omega*tcoord;
%-------------------------------------------
%        Plot predicted angular position vs time
%-------------------------------------------
figure (1)
subplot(2,1,1)
plot(tcoord,angposp5,'k')
xlabel('Time (sec)')
title ('1000 RPM, 100 ft*lbf Phase Deviation')
ylabel('Flywheel (Theta five) (rad)')
grid
```

```
subplot(2,1,2)
plot(tcoord,angposp1,'k')
xlabel('Time (sec)')
ylabel('Crankshaft nose (Theta one) (rad)')
grid
legend ('Meas     ','Pred     ')
%-------------------------------------------------
%         Compare Other degrees of freedom to theta 5
%-------------------------------------------------
twist = theta - ([1;1;1;1;1;1]*theta(5,:));
figure (3)
plot (tcoord,twist(1,:),'y'), hold on
plot (tcoord,twist(2,:),'r')
plot (tcoord,twist(3,:),'g')
plot (tcoord,twist(4,:),'b')
plot (tcoord,twist(6,:),'c')
title('Angular Deviation from Theta Five (Flywheel)')
xlabel('Time (sec)')
ylabel('Angular Deviation from Theta Five (radians)')
legend ('O.E.','Cyl #1','Cyl #2','Cyl #3','Dyno','Measured')
grid
orient landscape
%-------------------------------------------------
%         Plot Measured vs. Predicted Crankshaft Twist
%-------------------------------------------------
thetar1 = interp1(time1,pos1,tcoord,'spline');
thetar5 = interp1(time5,pos5,tcoord,'spline');
figure (6)
plot (tcoord,thetar1-thetar5,'k.')
grid, hold on
plot (tcoord,twist(2,:),'k')
title ('1000 RPM 100 ft*lbf Crankshaft Twist')
xlabel ('Time (sec)')
ylabel ('Twist (radians)')
legend('Meas     ','Pred     ')

function [k] = kelm(tl,tr,i)
%
% KELM calculates the element matrix k,
% given tl and tr as inputs
% Global variables
global j2rec j3rec j4rec
%-------------
% Constants
%-------------
h = tr-tl;
j1 = 0.02443;      %lbf*in*sec^2/rad
j2 = 0.2482;       %lbf*in*sec^2/rad
j3 = 0.1462;       %lbf*in*sec^2/rad
j4 = 0.2482;       %lbf*in*sec^2/rad
j5 = 7.222;                %lbf*in*sec^2/rad
j6 = 0.2870;       %lbf*in*sec^2/rad
k1 = 3.11e6;       %lbf*in/rad
k2 = 7.00e6;       %lbf*in/rad
k3 = 7.00e6;       %lbf*in/rad
```

```
k4 = 10.82e6;     %lbf*in/rad
k5 = 1.304e6;     %lbf*in/rad
c12 = 0.01;               %lbf*in*sec/rad
c23 = 0.01;               %lbf*in*sec/rad
c34 = 0.01;               %lbf*in*sec/rad
c45 = 0.01;               %lbf*in*sec/rad
c56 = 0.01;               %lbf*in*sec/rad
c2 = 0.013;               %lbf*in*sec/rad
c3 = 0.013;               %lbf*in*sec/rad
c4 = 0.013;               %lbf*in*sec/rad
%--------------------
k = zeros (12,12);
%--------------------
k(1,1)=1/6*(2*k1*tr^3-2*k1*tl^3-3*c12*tr^2-3*c12*tl^2+6*k1*tr*tl^2-6*j1*tr+6*...
  j1*tl+6*c12*tr*tl-6*k1*tr^2*tl)/h^2;
k(1,2)=1/6*(-2*k1*tr^3+2*k1*tl^3+3*c12*tr^2+3*c12*tl^2-6*k1*tr*tl^2-6*c12*tr*tl+...
  6*k1*tr^2*tl)/h^2;
k(1,7)=-1/6*(-k1*tr^3+k1*tl^3-3*c12*tr^2-3*c12*tl^2-3*k1*tr*tl^2+3*k1*tr^2*tl-6*...
  j1*tr+6*j1*tl+6*c12*tr*tl)/h^2;
k(1,8)=-1/6*(k1*tr^3-k1*tl^3+3*c12*tr^2+3*c12*tl^2+3*k1*tr*tl^2-3*k1*tr^2*tl-6*...
  c12*tr*tl)/h^2;
k(2,1)=1/6*(-2*k1*tr^3+2*k1*tl^3+3*c12*tr^2+3*c12*tl^2-6*k1*tr*tl^2-6*c12*tr*tl+...
  6*k1*tr^2*tl)/h^2;
k(2,2)=1/6*(6*k2*tr*tl^2-3*c2*tr^2-3*c2*tl^2-6*j2rec(i)*tr+6*j2rec(i)*tl-6*j2*tr+...
  6*j2*tl+6*tl*tr*c23+6*tr*c2*tl-6*k2*tl*tr^2+6*c12*tr*tl-6*k1*tr^2*tl+2*...
  k2*tr^3-2*k2*tl^3-3*c23*tr^2-3*c23*tl^2+2*k1*tr^3-2*k1*tl^3-3*c12*tr^2-...
  3*c12*tl^2+6*k1*tr*tl^2)/h^2;
k(2,3)=1/6*(-2*k2*tr^3+2*k2*tl^3-6*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2-6*tl*tr*c23+...
  6*k2*tl*tr^2)/h^2;
k(2,7)=-1/6*(k1*tr^3-k1*tl^3+3*c12*tr^2+3*c12*tl^2+3*k1*tr*tl^2-3*k1*tr^2*tl-6*...
  c12*tr*tl)/h^2;
k(2,8)=-1/6*(-3*k2*tr*tl^2-3*c2*tr^2-3*c2*tl^2-6*j2rec(i)*tr+6*j2rec(i)*tl-6*j2*...
  tr+6*j2*tl+6*tl*tr*c23+6*tr*c2*tl+3*k2*tl*tr^2+6*c12*tr*tl+3*k1*tr^2*...
  tl-k2*tr^3+k2*tl^3-3*c23*tr^2-3*c23*tl^2-k1*tr^3+k1*tl^3-3*c12*tr^2-3*...
  c12*tl^2-3*k1*tr*tl^2)/h^2;
k(2,9)=-1/6*(k2*tr^3-k2*tl^3+3*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2-3*k2*tl*tr^2-6*...
  tl*tr*c23)/h^2;
k(3,2)=1/6*(-2*k2*tr^3+2*k2*tl^3-6*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2-6*tl*tr*c23+...
  6*k2*tl*tr^2)/h^2;
k(3,3)=1/6*(6*k2*tr*tl^2+6*tl*tr*c23-6*k2*tl*tr^2+6*k3*tr*tl^2-6*k3*tr^2*tl+6*...
  tr*tl*c34+2*k2*tr^3-2*k2*tl^3-3*c23*tr^2-3*c23*tl^2+2*k3*tr^3-2*k3*...
  tl^3-3*c34*tr^2-3*c34*tl^2-3*c3*tr^2-3*c3*tl^2+6*tl*c3*tr-6*j3*tr+6*...
  j3*tl-6*j3rec(i)*tr+6*j3rec(i)*tl)/h^2;
k(3,4)=1/6*(-2*k3*tr^3+2*k3*tl^3-6*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2-6*tr*tl*c34+...
  6*k3*tr^2*tl)/h^2;
k(3,8)=-1/6*(k2*tr^3-k2*tl^3+3*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2-3*k2*tl*tr^2-6*...
  tl*tr*c23)/h^2;
k(3,9)=-1/6*(-3*k2*tr*tl^2+6*tl*tr*c23+3*k2*tl*tr^2-3*k3*tr*tl^2+3*k3*tr^2*tl+6*...
  tr*tl*c34-k2*tr^3+k2*tl^3-3*c23*tr^2-3*c23*tl^2-k3*tr^3+k3*tl^3-3*c34*...
  tr^2-3*c34*tl^2-3*c3*tr^2-3*c3*tl^2+6*tl*c3*tr-6*j3*tr+6*j3*tl-6*...
  j3rec(i)*tr+6*j3rec(i)*tl)/h^2;
k(3,10)=-1/6*(k3*tr^3-k3*tl^3+3*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2-3*k3*tr^2*tl-...
  6*tr*tl*c34)/h^2;
k(4,3)=1/6*(-2*k3*tr^3+2*k3*tl^3-6*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2-6*tr*tl*c34+...
  6*k3*tr^2*tl)/h^2;
```

```
k(4,4)=1/6*(6*tr*c4*tl+6*tr*c45*tl+6*k3*tr*tl^2-6*k3*tr^2*tl+6*tr*tl*c34+2*k3*...
  tr^3-2*k3*tl^3-3*c34*tr^2-3*c34*tl^2+6*tr*k4*tl^2-6*k4*tl*tr^2+2*k4*...
  tr^3-2*k4*tl^3-3*c45*tr^2-3*c45*tl^2-3*c4*tr^2-3*c4*tl^2-6*j4*tr+6*...
  j4*tl-6*j4rec(i)*tr+6*j4rec(i)*tl)/h^2;
k(4,5)=1/6*(-2*k4*tr^3+2*k4*tl^3-6*tr*k4*tl^2+3*c45*tr^2+3*c45*tl^2-6*tr*c45*...
  tl+6*k4*tl*tr^2)/h^2;
k(4,9)=-1/6*(k3*tr^3-k3*tl^3+3*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2-3*k3*tr^2*tl-6*...
  tr*tl*c34)/h^2;
k(4,10)=-1/6*(6*tr*c4*tl+6*tr*c45*tl-3*k3*tr*tl^2+3*k3*tr^2*tl+6*tr*tl*c34-k3*...
  tr^3+k3*tl^3-3*c34*tr^2-3*c34*tl^2-3*tr*k4*tl^2+3*k4*tl*tr^2-k4*tr^3+...
  k4*tl^3-3*c45*tr^2-3*c45*tl^2-3*c4*tr^2-3*c4*tl^2-6*j4*tr+6*j4*tl-6*...
  j4rec(i)*tr+6*j4rec(i)*tl)/h^2;
k(4,11)=-1/6*(k4*tr^3-k4*tl^3-3*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2+3*tr*k4*tl^2-6*...
  tr*c45*tl)/h^2;
k(5,4)=1/6*(-2*k4*tr^3+2*k4*tl^3-6*tr*k4*tl^2+3*c45*tr^2+3*c45*tl^2-6*tr*c45*tl+...
  6*k4*tl*tr^2)/h^2;
k(5,5)=1/6*(2*k4*tr^3-2*k4*tl^3+2*k5*tr^3-2*k5*tl^3-3*c45*tr^2-3*c45*tl^2-3*c56*...
  tr^2-3*c56*tl^2+6*tr*k4*tl^2+6*k5*tr*tl^2-6*j5*tr+6*j5*tl+6*tr*c45*tl+...
  6*tr*c56*tl-6*k4*tl*tr^2-6*k5*tr^2*tl)/h^2;
k(5,6)=1/6*(-2*k5*tr^3+2*k5*tl^3-6*k5*tr*tl^2+3*c56*tr^2+3*c56*tl^2-6*tr*c56*tl+...
  6*k5*tr^2*tl)/h^2;
k(5,10)=-1/6*(k4*tr^3-k4*tl^3-3*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2+3*tr*k4*tl^2-6*...
  tr*c45*tl)/h^2;
k(5,11)=-1/6*(6*tr*c45*tl-3*k5*tr*tl^2+3*k5*tr^2*tl+6*tr*c56*tl-k5*tr^3+k5*tl^3-...
  3*c56*tr^2-3*c56*tl^2-3*tr*k4*tl^2+3*k4*tl*tr^2-k4*tr^3+k4*tl^3-3*c45*...
  tr^2-3*c45*tl^2-6*j5*tr+6*j5*tl)/h^2;
k(5,12)=-1/6*(k5*tr^3-k5*tl^3+3*c56*tr^2+3*c56*tl^2+3*k5*tr*tl^2-3*k5*tr^2*tl-6*...
  tr*c56*tl)/h^2;
k(6,5)=1/6*(-2*k5*tr^3+2*k5*tl^3-6*k5*tr*tl^2+3*c56*tr^2+3*c56*tl^2-6*tr*c56*tl+...
  6*k5*tr^2*tl)/h^2;
k(6,6)=1/6*(2*k5*tr^3-2*k5*tl^3+6*k5*tr*tl^2-3*c56*tr^2-3*c56*tl^2-6*j6*tr+6*j6*...
  tl+6*tr*c56*tl-6*k5*tr^2*tl)/h^2;
k(6,11)=-1/6*(k5*tr^3-k5*tl^3+3*c56*tr^2+3*c56*tl^2+3*k5*tr*tl^2-3*k5*tr^2*tl-...
  6*tr*c56*tl)/h^2;
k(6,12)=-1/6*(-k5*tr^3+k5*tl^3-3*c56*tr^2-3*c56*tl^2-3*k5*tr*tl^2+3*k5*tr^2*tl-...
  6*j6*tr+6*j6*tl+6*tr*c56*tl)/h^2;
k(7,1)=-1/6*(-k1*tr^3+k1*tl^3+3*c12*tr^2+3*c12*tl^2-3*k1*tr*tl^2+3*k1*tr^2*tl-...
  6*j1*tr+6*j1*tl-6*c12*tr*tl)/h^2;
k(7,2)=1/6*(-k1*tr^3+k1*tl^3+3*c12*tr^2+3*c12*tl^2-3*k1*tr*tl^2+3*k1*tr^2*tl-...
  6*c12*tr*tl)/h^2;
k(7,7)=1/6*(2*k1*tr^3-2*k1*tl^3-6*k1*tr^2*tl+3*c12*tr^2+3*c12*tl^2-6*j1*tr+6*...
  j1*tl+6*k1*tr*tl^2-6*c12*tr*tl)/h^2;
k(7,8)=-1/6*(2*k1*tr^3-2*k1*tl^3-6*k1*tr^2*tl+3*c12*tr^2+3*c12*tl^2-6*c12*tr*...
  tl+6*k1*tr*tl^2)/h^2;
k(8,1)=1/6*(-k1*tr^3+k1*tl^3+3*c12*tr^2+3*c12*tl^2-3*k1*tr*tl^2+3*k1*tr^2*tl-...
  6*c12*tr*tl)/h^2;
k(8,2)=-1/6*(-3*k2*tr*tl^2+3*c2*tr^2+3*c2*tl^2-6*j2rec(i)*tr+6*j2rec(i)*tl-6*j2*...
  tr+6*j2*tl-6*tl*tr*c23-6*tr*c2*tl+3*k2*tl*tr^2-6*c12*tr*tl+3*k1*tr^2*...
  tl-k2*tr^3+k2*tl^3+3*c23*tr^2+3*c23*tl^2-k1*tr^3+k1*tl^3+3*c12*tr^2+...
  3*c12*tl^2-3*k1*tr*tl^2)/h^2;
k(8,3)=1/6*(-k2*tr^3+k2*tl^3-3*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2+3*k2*tl*tr^2-...
  6*tl*tr*c23)/h^2;
k(8,7)=-1/6*(2*k1*tr^3-2*k1*tl^3-6*k1*tr^2*tl+3*c12*tr^2+3*c12*tl^2-6*c12*tr*...
  tl+6*k1*tr*tl^2)/h^2;
k(8,8)=1/6*(6*k2*tr*tl^2+3*c2*tr^2+3*c2*tl^2-6*j2rec(i)*tr+6*j2rec(i)*tl-6*j2*...
```

```
  tr+6*j2*tl-6*tl*tr*c23-6*tr*c2*tl-6*k2*tl*tr^2-6*c12*tr*tl-6*k1*tr^2*...
  tl+2*k2*tr^3-2*k2*tl^3+3*c23*tr^2+3*c23*tl^2+2*k1*tr^3-2*k1*tl^3+3*...
  c12*tr^2+3*c12*tl^2+6*k1*tr*tl^2)/h^2;
k(8,9)=-1/6*(2*k2*tr^3-2*k2*tl^3-6*k2*tl*tr^2+3*c23*tr^2+3*c23*tl^2-6*tl*tr*...
  c23+6*k2*tr*tl^2)/h^2;
k(9,2)=1/6*(-k2*tr^3+k2*tl^3-3*k2*tr*tl^2+3*c23*tr^2+3*c23*tl^2+3*k2*tl*tr^2-...
  6*tl*tr*c23)/h^2;
k(9,3)=-1/6*(-3*k2*tr*tl^2-6*tl*tr*c23+3*k2*tl*tr^2-3*k3*tr*tl^2+3*k3*tr^2*tl-...
  6*tr*tl*c34-k2*tr^3+k2*tl^3+3*c23*tr^2+3*c23*tl^2-k3*tr^3+k3*tl^3+...
  3*c34*tr^2+3*c34*tl^2+3*c3*tr^2+3*c3*tl^2-6*tl*c3*tr-6*j3*tr+6*j3*tl-...
  6*j3rec(i)*tr+6*j3rec(i)*tl)/h^2;
k(9,4)=1/6*(-k3*tr^3+k3*tl^3-3*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2+3*k3*tr^2*tl-...
  6*tr*tl*c34)/h^2;
k(9,8)=-1/6*(2*k2*tr^3-2*k2*tl^3-6*k2*tl*tr^2+3*c23*tr^2+3*c23*tl^2-6*tl*tr*...
  c23+6*k2*tr*tl^2)/h^2;
k(9,9)=1/6*(6*k2*tr*tl^2-6*tl*tr*c23-6*k2*tl*tr^2+6*k3*tr*tl^2-6*k3*tr^2*tl-...
  6*tr*tl*c34+2*k2*tr^3-2*k2*tl^3+3*c23*tr^2+3*c23*tl^2+2*k3*tr^3-...
  2*k3*tl^3+3*c34*tr^2+3*c34*tl^2+3*c3*tr^2+3*c3*tl^2-6*tl*c3*tr-...
  6*j3*tr+6*j3*tl-6*j3rec(i)*tr+6*j3rec(i)*tl)/h^2;
k(9,10)=-1/6*(2*k3*tr^3-2*k3*tl^3-6*k3*tr^2*tl+3*c34*tr^2+3*c34*tl^2-6*tr*...
  tl*c34+6*k3*tr*tl^2)/h^2;
k(10,3)=1/6*(-k3*tr^3+k3*tl^3-3*k3*tr*tl^2+3*c34*tr^2+3*c34*tl^2+3*k3*tr^2*tl-...
  6*tr*tl*c34)/h^2;
k(10,4)=-1/6*(-6*tr*c4*tl-6*tr*c45*tl-3*k3*tr*tl^2+3*k3*tr^2*tl-6*tr*tl*c34-...
  k3*tr^3+k3*tl^3+3*c34*tr^2+3*c34*tl^2-3*tr*k4*tl^2+3*k4*tl*tr^2-...
  k4*tr^3+k4*tl^3+3*c45*tr^2+3*c45*tl^2+3*c4*tr^2+3*c4*tl^2-6*j4*tr+...
  6*j4*tl-6*j4rec(i)*tr+6*j4rec(i)*tl)/h^2;
k(10,5)=1/6*(-k4*tr^3+k4*tl^3+3*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2-3*tr*k4*tl^2-...
  6*tr*c45*tl)/h^2;
k(10,9)=-1/6*(2*k3*tr^3-2*k3*tl^3-6*k3*tr^2*tl+3*c34*tr^2+3*c34*tl^2-6*tr*tl*...
  c34+6*k3*tr*tl^2)/h^2;
k(10,10)=1/6*(-6*tr*c4*tl-6*tr*c45*tl+6*k3*tr*tl^2-6*k3*tr^2*tl-6*tr*tl*c34+2*...
  k3*tr^3-2*k3*tl^3+3*c34*tr^2+3*c34*tl^2+6*tr*k4*tl^2-6*k4*tl*tr^2+...
  2*k4*tr^3-2*k4*tl^3+3*c45*tr^2+3*c45*tl^2+3*c4*tr^2+3*c4*tl^2-6*j4*...
  tr+6*j4*tl-6*j4rec(i)*tr+6*j4rec(i)*tl)/h^2;
k(10,11)=-1/6*(2*k4*tr^3-2*k4*tl^3-6*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2-6*tr*c45*...
  tl+6*tr*k4*tl^2)/h^2;
k(11,4)=1/6*(-k4*tr^3+k4*tl^3+3*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2-3*tr*k4*tl^2-...
  6*tr*c45*tl)/h^2;
k(11,5)=-1/6*(-6*tr*c45*tl-3*k5*tr*tl^2+3*k5*tr^2*tl-6*tr*c56*tl-k5*tr^3+k5*tl^3+...
  3*c56*tr^2+3*c56*tl^2-3*tr*k4*tl^2+3*k4*tl*tr^2-k4*tr^3+k4*tl^3+3*c45*...
  tr^2+3*c45*tl^2-6*j5*tr+6*j5*tl)/h^2;
k(11,6)=1/6*(-k5*tr^3+k5*tl^3+3*c56*tr^2+3*c56*tl^2-3*k5*tr*tl^2+3*k5*tr^2*tl-...
  6*tr*c56*tl)/h^2;
k(11,10)=-1/6*(2*k4*tr^3-2*k4*tl^3-6*k4*tl*tr^2+3*c45*tr^2+3*c45*tl^2-6*tr*c45*tl+...
  6*tr*k4*tl^2)/h^2;
k(11,11)=1/6*(2*k4*tr^3-2*k4*tl^3+2*k5*tr^3-2*k5*tl^3+3*c45*tr^2+3*c45*tl^2+3*c56*...
  tr^2+3*c56*tl^2-6*k4*tl*tr^2-6*k5*tr^2*tl-6*j5*tr+6*j5*tl+6*tr*k4*tl^2+...
  6*k5*tr*tl^2-6*tr*c45*tl-6*tr*c56*tl)/h^2;
k(11,12)=-1/6*(2*k5*tr^3-2*k5*tl^3-6*k5*tr^2*tl+3*c56*tr^2+3*c56*tl^2-6*tr*c56*tl+...
  6*k5*tr*tl^2)/h^2;
k(12,5)=1/6*(-k5*tr^3+k5*tl^3+3*c56*tr^2+3*c56*tl^2-3*k5*tr*tl^2+3*k5*tr^2*tl-...
  6*tr*c56*tl)/h^2;
k(12,6)=-1/6*(-k5*tr^3+k5*tl^3+3*c56*tr^2+3*c56*tl^2-3*k5*tr*tl^2+3*k5*tr^2*tl-...
  6*j6*tr+6*j6*tl-6*tr*c56*tl)/h^2;
```

```
k(12,11)=-1/6*(2*k5*tr^3-2*k5*tl^3-6*k5*tr^2*tl+3*c56*tr^2+3*c56*tl^2-6*tr*c56*tl+...
  6*k5*tr*tl^2)/h^2;
k(12,12)=1/6*(2*k5*tr^3-2*k5*tl^3-6*k5*tr^2*tl+3*c56*tr^2+3*c56*tl^2-6*j6*tr+6*j6*...
  tl+6*k5*tr*tl^2-6*tr*c56*tl)/h^2;
```

Programs used to calculate cylinder gas torques, given angular position data at θ_1 and θ_5.

```
%       SPRESSF2
%-------------------------------------------------------------
%       Program to determine cylinder pressures in 3 cylinder
%       two stroke diesel engine, given instantaneous angular
%       velocity at two of the six degrees of freedom
%       (crankshaft nose and flywheel)
%-------------------------------------------------------------
ic = [0 0 0 -1.2 -1.2 -10.4]*1e-04;          % set initial conditions
%------------
%  load data
%------------
load w51k100.md
t = [0;w51k100(1:7938,1)];
tt = diff (t);                               % COMPUTES THE DEL_T'S
tt = reshape(tt,126,63);                     % This phase locks one cycle
mdt = mean(tt);                              % Mean dt for each cycle
for i = 1:63;                                % Reference each cycle to its own mean
  tt(:,i) = tt(:,i)./mean(tt(:,i));
end
tt = tt'*mean(mdt);
dtrat = tt(1)/mean(tt(2:63,1));
teeth = linspace(0,2*pi,127); tooth = 2*pi/127;
th51 = ic(5) + tooth*dtrat;
tttt = mean(tt);                             % This ensemble averages the phases
cyctm = sum(tttt);                           % Time for one complete cycle
thetar5 = [ic(5) teeth(1:125)+th51];
thetar5(127) = 2*pi+ic(5);
time5 = [0,cumsum(tttt)-(1-dtrat)*tttt(1)];
time5(127) = cyctm;
load w11k100.md
t =  [0;w11k100(1:7920,1)];
tt = diff(t);
tt = reshape(tt,720,11)';                    % Phase lock one cycle
tt = tt(2:11,:);
tttt = mean(tt);                             % Ensemble average the phases
thetar1 = linspace(0,(2*pi),721);            % Known positions of the O.E. windows
time1 = [0 cumsum(tttt)];
time1 = time1*(cyctm/max(time1));            % Adjust times to agree
omega = (1/cyctm)*2*pi;
N = 512;                                     % set number of nodes for solution
%-------------------------------------------------
%  calculate, interpolate, and filter position data
%-------------------------------------------------
time = linspace(0,cyctm,N)';
dt = cyctm/N;
theta = linspace(0,2*pi,N)';
```

```
fcv = 100*dt;      % filtering cut-off value as ratio of sampling freq
thetar1 = interp1(time1,thetar1,time,'spline');
thetar5 = interp1(time5,thetar5,time,'spline');
theta1 = theta + vfilt(thetar1-theta,fcv);
theta5 = theta + vfilt(thetar5-theta,fcv);
%---------------------------------------
%  derive velocity and acceleration data
%---------------------------------------
omega1 = deriv(theta1,dt);
omega5 = deriv(theta5,dt);
accel1 = deriv(omega1,dt);
accel5 = deriv(omega5,dt);
%-----------------------------------------------
%  plot raw and filtered omegas for the two dofs
%-----------------------------------------------
figure (1)
plot (time, deriv(thetar1,dt), 'mX'), grid, hold on
plot (time, omega1, 'r')
plot (time, deriv(thetar5,dt), 'cX')
plot (time, omega5, 'b')
plot (time, ones(size(omega5))*omega,'g')
title ('Raw and Filtered Angular Velocity at DOFS 1 and 5')
xlabel ('time (seconds)')
ylabel ('angular velocity (rad/sec)')
orient landscape
%-----------------------------------------------
%  plot raw and filtered thetas for the two dofs
%-----------------------------------------------
figure (2)
plot (time, thetar1-theta, 'mX'), grid, hold on
plot (time, theta1-theta, 'r')
plot (time, thetar5-theta, 'cX')
plot (time, theta5-theta, 'b')
title ('Raw and Filtered phase deviation at DOFS 1 and 5')
xlabel ('time (seconds)')
ylabel ('phase deviation from mean (rad)')
orient landscape
%------------------------
%  set calibration data
%------------------------
j1=0.02443;       %lb*in*sec^2/rad
j2=0.2482;        %lb*in*sec^2/rad
j3=0.1462;        %lb*in*sec^2/rad
j4=0.2482;        %lb*in*sec^2/rad
j5=7.222;         %lb*in*sec^2/rad
j6=0.2870;        %lb*in*sec^2/rad
k1=3.11e6;        %lb*in/rad
k2=7.00e6;        %lb*in/rad
k3=7.00e6;        %lb*in/rad
k4=10.82e6;       %lb*in/rad
k5=1.304e6;       %lb*in/rad
c12=0.01;         %lb*in*sec/rad
c23=0.01;         %lb*in*sec/rad
c34=0.01;         %lb*in*sec/rad
c45=0.01;         %lb*in*sec/rad
```

```
c56=0.01;          %lb*in*sec/rad
c2=0.013;          %lb*in*sec/rad
c3=0.013;          %lb*in*sec/rad
c4=0.013;          %lb*in*sec/rad
Tload = 1200;      % lbf*in Load torque
Fpar = 100;        % lbf              Parasitic force
Taux = 168;        % lbf*in Valvetrain and auxilliary torque
Tpmp = 9.5;        % lbf*in Oil pump torque
%------------------------------
%  Solve equations 6, 5, and 1
%------------------------------
global AA BB CC DD TT
icomega = 105.5*[1 1 1 1 1 1];
thetar2=zeros(N,1); thetar4=thetar2; thetar6=thetar2; thetar3=thetar2;
omega2=zeros(N,1); omegar3=omega2; omega4=omega2; omegar6=omega2;
thetar3(1) = ic(3);
omegar3(1) = icomega(3);
%------ solve equation 6 for theta6 and omega6------
bcval = [ic(6),ic(6)+2*pi];
AA = j6;  BB = c56;  CC = k5;
DD = -Tload + c56*omega5 +k5*theta5;
ff = zeros(N,1);
kk = zeros(N,N);
for i = 1:(N-1);
  h = time(i+1)-time(i);
  k = -(AA/h)*[1 -1;-1 1] + (BB/2)*[-1 1;-1 1] + (CC*h/6)*[2 1;1 2];
  f = (h/2)*[DD(i);DD(i+1)];
  kk(i:i+1,i:i+1) = kk(i:i+1,i:i+1) + k;
  ff(i:i+1) = ff(i:i+1) + f;
end
%----- apply boundary conditions -----
kk(1,:) = zeros(1,N);  kk(N,:) = zeros(1,N);
kk(1,1) = 1;  kk(N,N) = 1;
ff(1) = bcval(1);  ff(N) = bcval(2);
%----- solve matrix eqn -----
theta6 = kk\ff;
omega6 = deriv(theta6,dt);
%------ solve equation 5 for theta4 ------
thetar4 = (Taux + k5*(theta5-theta6) + c56*(omega5-omega6) +...
      k4*theta5 + j5*accel5)./k4;
theta4 = theta + vfilt(thetar4-theta,fcv);
%------ solve equation 1  for theta2 ------
thetar2 = (Tpmp + k1*thetar1 + j1*accel1)./k1;
theta2 = theta + vfilt(thetar2-theta,fcv);
%------ compute omega and accel for dofs 2 and 4 ------
omega2 = deriv(theta2,dt);          % raw velocities
omega4 = deriv(theta4,dt);
accel2 = deriv(omega2,dt);                  % calculated accelerations
accel4 = deriv(omega4,dt);
%---------------------------------------------
%  Solve equations 2, 3, and 4 in three steps
%---------------------------------------------
R = 2.25;          % Crankshaft eccentricity (in)
W = 7.556;         % reciprocating weight(lbf)
g = 386;           % The acceleration of gravity (in/sec^2)
```

```
B = 3.875;          % Cylinder Bore (in)
L = 8.80;           % Connecting Rod length (in)
j2rec=(W*R^2/(2*g))*(1-cos(2*theta));                %lb*in*sec^2
j3rec=(W*R^2/(2*g))*(1-cos(2*(theta-4*pi/3)));       %lb*in*sec^2
j4rec=(W*R^2/(2*g))*(1-cos(2*(theta-2*pi/3)));       %lb*in*sec^2
T1cyl = zeros(N,1);  T2cyl=T1cyl;  T3cyl = T1cyl;
pref = 14.706 + 0.00205*((omega*60/(2*pi))-634);  % ref press (psia)
N1 = min(find(theta1>=(2*pi/3)));                 % index for TDC Cyl #3
N2 = min(find(theta1>=(4*pi/3)));                 % index for TDC Cyl #2
accel3 = zeros(size(accel2));
s1 = R*cos(theta) + sqrt(L^2 - (R^2)*sin(theta).^2);
s2 = R*cos(theta-4*pi/3) + sqrt(L^2 - (R^2)*sin(theta-4*pi/3).^2);
s3 = R*cos(theta-2*pi/3) + sqrt(L^2 - (R^2)*sin(theta-2*pi/3).^2);
Sp1 = deriv(s1,dt);  Spd1 = deriv(Sp1,dt);        % Piston speed (in/sec) and
Sp2 = deriv(s2,dt);  Spd2 = deriv(Sp2,dt);        % Piston acceleration (in/sec^2)
Sp3 = deriv(s3,dt);  Spd3 = deriv(Sp3,dt);
T1rec = -(W/g)*Spd1.*(Sp1/omega);                 % in*lbf Reciprocating torque
T2rec = -(W/g)*Spd2.*(Sp2/omega);
T3rec = -(W/g)*Spd3.*(Sp3/omega);
Tpar1 = Fpar*abs(Sp1/omega);                      % in*lbf Parasitic torque
Tpar2 = Fpar*abs(Sp2/omega);
Tpar3 = Fpar*abs(Sp3/omega);
%------ Step one: Determine known values of Tcyl from pref ------
%------ (known values of Tcyl are 0) ------
%------ Step two: Solve for theta3 throughout cycle ------
%------ solve equation 2 for theta3 ------
thetar3(N1:(N2-1)) = ((j2+j2rec(N1:(N2-1))).*accel2(N1:(N2-1)) + ...
    c12*(omega2(N1:(N2-1))-omega1(N1:(N2-1))) + ...
    k1*(theta2(N1:(N2-1))-theta1(N1:(N2-1))) + k2*theta2(N1:(N2-1)) + ...
    c2*omega2(N1:(N2-1)) - T1cyl(N1:(N2-1)) - T1rec(N1:(N2-1)) + ...
    Tpar1(N1:(N2-1)))./k2;
%------ solve equation 3 for theta3 and omega3 ------
BB = c23+c34+c3;  CC = k2+k3;
DDT = T2cyl + T2rec - Tpar2 + c23*omega2 + k2*theta2 + c34*omega4 + k3*theta4;
for i = 1:(N1-2);
  AA = j3 + j3rec(i:i+1);  TT = time(i:i+1);  DD = DDT(i:i+1);
  [T,X] = ode45('seqns2',[time(i),time(i+1)],[thetar3(i);omegar3(i)]);
  thetar3(i+1) = X(length(T),1);
  omegar3(i+1) = X(length(T),2);
end
%------ solve equation 4 for theta3 ------
thetar3(N2:N) = ((j4+j4rec(N2:N)).*accel4(N2:N) + k3*theta4(N2:N) + ...
    (c45+c4)*omega4(N2:N) - c45*omega5(N2:N) + k4*(theta4(N2:N) - ...
    theta5(N2:N)) - T3cyl(N2:N) - T3rec(N2:N) + Tpar3(N2:N))./k3;
%------ filter theta3 and derive omega3 and accel3 ------
theta3 = theta + vfilt(thetar3-theta,fcv);
omega3 = deriv(theta3,dt);
accel3 = deriv(omega3,dt);
%------ Step three: solve for remaining Tcyl values ------
%------ solve equation 2 for T1cyl ------
T1cyl(1:(N1-1)) = -T1rec(1:(N1-1)) + Tpar1(1:(N1-1)) + (j2+j2rec(1:(N1-1))).*accel2(1:(N1-1)) + ...
  c12*(omega2(1:(N1-1))-omega1(1:(N1-1))) + k1*(theta2(1:(N1-1))-theta1(1:(N1-1))) +...
  c23*(omega2(1:(N1-1))-omega3(1:(N1-1))) + k2*(theta2(1:(N1-1))-theta3(1:(N1-1))) +...
  c2*omega2(1:(N1-1));
T1cyl(N2:N) = -T1rec(N2:N) + Tpar1(N2:N) + (j2+j2rec(N2:N)).*accel2(N2:N) + ...
```

```
  c12*(omega2(N2:N)-omega1(N2:N)) + k1*(theta2(N2:N)-theta1(N2:N)) + ...
  c23*(omega2(N2:N)-omega3(N2:N)) + k2*(theta2(N2:N)-theta3(N2:N)) + ...
  c2*omega2(N2:N);
%------ solve equation 3 for T2cyl ------
T2cyl(N1:N) = -T2rec(N1:N) + Tpar2(N1:N) + (j3+j3rec(N1:N)).*accel3(N1:N) + ...
  c23*(omega3(N1:N)-omega2(N1:N)) + k2*(theta3(N1:N)-theta2(N1:N)) + ...
  c34*(omega3(N1:N)-omega4(N1:N)) + k3*(theta3(N1:N)-theta4(N1:N)) + ...
  c3*omega3(N1:N);
%------ solve equation 4 for T3cyl ------
T3cyl(1:(N2-1)) = -T3rec(1:(N2-1)) + Tpar3(1:(N2-1)) + (j4+j4rec(1:(N2-1))).*accel4(1:(N2-1)) + ...
  c34*(omega4(1:(N2-1))-omega3(1:(N2-1))) + k3*(theta4(1:(N2-1))-theta3(1:(N2-1))) +...
  c45*(omega4(1:(N2-1))-omega5(1:(N2-1))) + k4*(theta4(1:(N2-1))-theta5(1:(N2-1))) +...
  c4*omega4(1:(N2-1));
%------ Convert to FT*LBF ------
T1cyl = T1cyl./12;
T2cyl = T2cyl./12;
T3cyl = T3cyl./12;
%-----------------------------------------------------------
%  Determine and plot measured/predicted cylinder torques
%-----------------------------------------------------------
load wa1k100.md          % ECA cylinder #1 pressure data
load wb1k100.md          % ECA cylinder #2 pressure data
load wc1k100.md          % ECA cylinder #3 pressure data
pa = reshape (wa1k100,5,720); P1cyl = [pa(1,:) pa(1)];
pb = reshape (wb1k100,5,720); P2cyl = [pb(1,:) pb(1)];
pc = reshape (wc1k100,5,720); P3cyl = [pc(1,:) pc(1)];
pos = linspace(0,2*pi,721);
dtm = (2*pi/omega)/720;
s1m = R*cos(pos) + sqrt(L^2 - (R^2)*sin(pos).^2);
s2m = R*cos(pos-4*pi/3) + sqrt(L^2 - (R^2)*sin(pos-4*pi/3).^2);
s3m = R*cos(pos-2*pi/3) + sqrt(L^2 - (R^2)*sin(pos-2*pi/3).^2);
Sp1m = deriv(s1m,dtm);   % Piston speed (in/sec)
Sp2m = deriv(s2m,dtm);
Sp3m = deriv(s3m,dtm);
T1cylm = -((P1cyl.*pref-pref)*(pi*B^2/4).*Sp1m/omega)/12; % (ft*lbf)
T2cylm = -((P2cyl.*pref-pref)*(pi*B^2/4).*Sp2m/omega)/12; % (ft*lbf)
T3cylm = -((P3cyl.*pref-pref)*(pi*B^2/4).*Sp3m/omega)/12; % (ft*lbf)
figure(3)
plot(pos,T1cylm,'bX',pos,T2cylm,'bO',pos,T3cylm,'b.')
legend('Cyl #1','Cyl #2','Cyl #3')
grid, hold on
plot(theta1,T1cyl,'rX',theta1,T2cyl,'rO',theta1,T3cyl,'r.')
title('Measured (blue) and Predicted (red) Cylinder Gas Torques')
ylabel('Torque (ft*lbf)')
xlabel('Crank Angle')
orient landscape
figure (4)
plot(pos,T1cylm+T2cylm+T3cylm,'b')
grid, hold on
plot(theta1,T1cyl+T2cyl+T3cyl,'r')
title('Measured (blue) and Predicted (red) Total Cylinder Gas Torque')
ylabel ('Torque (ft*lbf)')
xlabel ('Crank Angle')
orient landscape
figure(5)
```

```
degm = 180*pos/pi;
degp = 180*theta/pi;
subplot(3,1,1)
plot(degm,T1cylm,'k.')
grid, hold on
plot(degp,T1cyl,'k')
legend ('Meas      ','Pred      ')
ylabel ('Cyl #1 Torque (ft*lbf)')
title('1000 RPM, 100 ft*lbf Cylinder Gas Torques')
axis([0,360,-500,1000])
subplot(3,1,2)
plot(degm,T2cylm,'k.')
grid, hold on
plot(degp,T2cyl,'k')
ylabel ('Cyl #2 Torque (ft*lbf)')
axis([0,360,-500,1000])
subplot(3,1,3)
plot(degm,T3cylm,'k.')
grid, hold on
plot(degp,T3cyl,'k')
ylabel ('Cyl#3 Torque (ft*lbf)')
xlabel('Crank Angle (deg)')
axis([0,360,-500,1000])
orient tall
figure (6)
plot(degm,T1cylm+T2cylm+T3cylm,'k.')
grid, hold on
plot(degp,T1cyl+T2cyl+T3cyl,'k')
title ('1000 RPM, 100 ft*lbf Total Cylinder Gas Torque')
legend ('Meas      ','Pred      ')
ylabel ('Total Cylinder Gas Torque (ft*lbf)')
xlabel ('Crank Angle (deg)')
axis([0,360,-400,800])
orient tall

%       DERIV
%       Function to determine 1-D derivative of a vector using
%       a central difference technique.
%       Xd = Deriv(X,dt)
%       Returns the derivative of the vector X as a function of
%       t, given the time step dt.  Default value for dt is 1.
function xd = deriv(x,dt)
if nargin == 1;
 dt = 1;
end
N = length(x);
xd = zeros(size(x));
xdf = diff(x);
xd(2:(N-1)) = (xdf(2:(N-1)) + xdf(1:(N-2)))./(2*dt);
xd(1) = xdf(1)/dt;
xd(N) = xdf(N-1)/dt;
%       Forward difference format
%xd(1:N-1) = xdf/dt;
%xd(N)=xd(1);
```

```
%       SEQNS2
%       Function to solve arbitrary 2nd order ode in form
%       AAxdd + BBxd + CCx = DD
%       where AA, BB, CC, and DD are global variables
function xdot=seqns2(t,x)
 global AA BB CC DD TT
DDS = DD(1) + diff(DD)*(t-TT(1))/diff(TT);
AAS = AA(1) + diff(AA)*(t-TT(1))/diff(TT);
xdot = zeros(2,1);
xdot(1) = x(2);
xdot(2) = (DDS - BB.*x(2) - CC.*x(1))./AAS;

%       VFILT
%       Function performs fast fourier transform filtration
%       of high frequency components of given data
%       Y = VFILT(X,FCV)
%       Y is filtered data
%       X is input data
%       FCV is frequency cutoff value (as a fraction of
%          the sampling frequency)
function Y = vfilt(x,fcv)
N = length(x);                          % number of data points
D = fft(x);                             % Fourier transform of data
ico = fix(fcv*N);                       % index in D
DF = zeros(1,N);
DF(1:(ico+1)) = D(1:(ico+1));           % filter high frequencies
DF((N-ico+1):N) = D((N-ico+1):N);       % mirror image data
Y = ifft(DF);                           % inverse fft
Y = real(Y);
Y = reshape(Y,size(x));
```

Program used for analysis of torsional model natural frequencies and modal shapes.

```
%----------------------------
%       NATFREQ
%----------------------------
%       Determines natural frequencies of torsional model
j1=0.02443;     %lb*in*sec^2/rad
j2=0.2482;      %lb*in*sec^2/rad
j3=0.1462;      %lb*in*sec^2/rad
j4=0.2482;      %lb*in*sec^2/rad
j5=7.222;       %lb*in*sec^2/rad
j6=0.2870;      %lb*in*sec^2/rad
jrec=0.02478;
k1=3.11e6;      %lb*in/rad
k2=7.00e6;      %lb*in/rad
k3=7.00e6;      %lb*in/rad
k4=10.82e6;     %lb*in/rad
k5=1.304e6;     %lb*in/rad
c12=0.01;       %lb*in*sec/rad
c23=0.01;       %lb*in*sec/rad
c34=0.01;       %lb*in*sec/rad
c45=0.01;       %lb*in*sec/rad
```

```
c56=0.01;          %lb*in*sec/rad
c2=0.013;          %lb*in*sec/rad
c3=0.013;          %lb*in*sec/rad
c4=0.013;          %lb*in*sec/rad
%-------------------------------------------------------------------------
J = [j1 0 0 0 0 0;0 j2+jrec 0 0 0 0;0 0 j3+jrec 0 0 0;...
     0 0 0 j4+jrec 0 0;0 0 0 0 j5 0;0 0 0 0 0 j6];
%-------------------------------------------------------------------------
K = [k1 -k1 0 0 0 0;-k1 k1+k2 -k2 0 0 0;0 -k2 k2+k3 -k3 0 0;...
     0 0 -k3 k3+k4 -k4 0;0 0 0 -k4 k4+k5 -k5;0 0 0 0 -k5 k5];
%-------------------------------------------------------------------------
ws = eig(K/J);
w = sqrt(ws)       % (rad/sec) natural frequencies
whz = w/(2*pi)     % (Hz) natural frequencies
for i = 1:6;
  kj = K-J*w(i)^2;
  A(1,i) = 1;
  A(2,i) = -A(1,i)*kj(1,1)/kj(1,2);
  A(3,i) = (-A(1,i)*kj(2,1)-A(2,i)*kj(2,2))/kj(2,3);
  A(4,i) = (-A(2,i)*kj(3,2)-A(3,i)*kj(3,3))/kj(3,4);
  A(5,i) = (-A(3,i)*kj(4,3)-A(4,i)*kj(4,4))/kj(4,5);
  A(6,i) = (-A(4,i)*kj(5,4)-A(5,i)*kj(5,5))/kj(5,6);
end
[whz,I] = sort(abs(whz));
A
figure (1)
for i = 2:6;
  subplot(5,1,i-1)
  plot(A(:,I(i)),'k'), grid, hold on
  plot(A(:,I(i)),'ko')
  ylabel ([num2str(whz(i),4),' Hz'])
  %title (['Mode ',num2str(i,1)])
end
orient tall
```

APPENDIX F. UNCERTAINTY ANALYSIS

Various sources of error will affect the accuracy of results in this study. The following discussion contains a qualitative summary of the potential sources of experimental uncertainty.

1. Optical Encoder

The use of the flexible coupling results in a torsional oscillation of the encoder disk about the actual angular position of the crankshaft nose (θ_1). The frequency of the oscillation is apparent from the measured data and discussed in Appendix C. Heidenhain [Ref 21] lists a kinematic error of transfer of ±10" for the encoder coupling, corresponding to a vibrational amplitude of about 1×10^{-4} radians. The oscillation seen in experimental data varies from about that value up to 1×10^{-3} radians. But this high frequency oscillation is easily filtered out of the raw data, and its amplitude is still an order of magnitude smaller than the amplitude of the crankshaft angular velocity fluctuations.

The 3600 count optical encoder allows a theoretical adjustment to within 0.1°, but this is only as accurate as the TDC alignment for the engine. Due to an inadequate method of determining TDC, this error may grow to 1 or 2°.

2. Flywheel

The TDC position for the flywheel data is determined by a comparison of the first Δt to subsequent values, assuming that the crankshaft has zero twist at TDC. This is a reasonable assumption, but not completely accurate. Therefore, an expected error is introduced due to ambiguity in determination of the flywheel TDC angular position.

Tooth to tooth variation will result in a cyclic error for the flywheel data. A motoring measurement of the flywheel at a constant speed could be used to generate a correction signal, eliminating this error [Ref 8]. This was not done for this study, assuming the error would be small relative to the measured velocity fluctuations.

A radial vibration is present in the flywheel during engine operation. Since the proximeter is mounted to view the ring gear teeth from the side, this radial vibration will cause the position of the proximeter relative to the tooth to oscillate up and down. This up and down oscillation will induce a cyclic variation in the pulse width not associated with crankshaft rotational velocity fluctuations. However, it is expected that this variation is reasonably small enough to be ignored.

3. Measurement Error

Engine speed and load vary slowly during data collection. Since data for the various runs are not collected simultaneously, there are small differences in speed and load for data comprising a set. Typically, engine speed variation was within 20 RPM of the stated value, and load was within 1 ft*lbf. A correction is made in the pressure prediction program to account for small differences in rotation speed between the θ_1 and θ_5 data.

LIST OF REFERENCES

1. Freestone, J. W., and E. G. Jenkins, "The Diagnosis of Cylinder Power Faults in Diesel Engines by Flywheel Speed Measurement," SAE paper 864747.

2. Mauer, Georg F., and Robert J.Watts, "On-Line Cylinder Diagnostics on Combustion Engines by Noncontact Torque and Speed Measurements," SAE paper 890485.

3. Mauer, Georg F., "On-Line Cylinder Fault Diagnostics for Internal Combustion Engines," IEEE Transactions on Industrial Electronics, Vol. 37, No. 3, June 1990.

4. Citron, Stephen J., John E. O'Higgins, and Lillian Y. Chen, "Cylinder by Cylinder Engine Pressure and Pressure Torque Waveform Determination Utilizing Speed Fluctuations," SAE paper 890486.

5. Connolly, Francis T., and Andrew E. Yagle, "Modeling and Identification of the Combustion Pressure Process in Internal Combustion Engines Using Engine Speed Fluctuations," Winter Annual Meeting of the ASME, Anaheim, CA, DSC-Vol. 44, Transportation Systems, 8-13 Nov 1992, pp. 191-206.

6. Connolly, Francis T., "Direct Estimation of Cyclic Combustion Pressure Variability Using Engine Speed Fluctuations in an Internal Combustion Engine," SAE paper 940143.

7. Lim, Byeongjin et al, "Estimation of the Cylinder Pressure in a SI Engine Using the Variation of Crankshaft Speed," SAE paper 940145.

8. Iida, Kazumasa, Katsuo Akishino, and Kazuo Kido, "IMEP Estimation from Instantaneous Crankshaft Torque Variation," SAE paper 900617.

9. Taraza, Dinu, "Estimation of the Mean Indicated Pressure from Measurements of the Crankshafts Angular Speed Variation," SAE paper 932413.

10. Rizzoni, Giorgio, "Estimate of Indicated Torque from Crankshaft Speed Fluctuations: A Model for the Dynamics of the IC Engine," IEEE Transactions on Vehicular Technology, Vol. 38, No. 3, August 1989.

11. Zhang, Yuxuan, and Giorgio Rizzoni, "An On-Line Indicated Torque Estimator for IC Engine Diagnosis and Control," DSC Advanced Automotive Technologies Proceedings of the 1993 ASME Winter Annual Meeting 28Nov-03Dec 1993 v 52 pp. 147-162.

12. Rizzoni, Giorgio, and Yuxuan Zhang, "Identification of a Nonlinear IC Engine Dynamic Model with Application to On-Line Indicated Torque Estimation," DSC Advanced Automotive Technologies Proceedings of the 1993 ASME Winter Annual Meeting 28Nov-03Dec 1993 v 52 pp. 199-209.

13. Srinivasan, Krishnaswami, Giorgio Rizzoni et al, "On-Line Estimation of Net Engine Torque from Crankshaft Angular Velocity Measurement using Repetitive Estimators," Proceedings of the 1992 American Control Conference, 24-26 Jun 1992 Chicago, IL, pp. 516-520.

14. Shiao, Yaojung, and John J. Moskwa, "Cylinder Pressure and Combustion Heat Release Estimation for SI Engine Diagnostics Using Nonlinear Sliding Observers," IEEE Transactions on Control Systems Technology, Vol. 3, No. 1, March 1995.

15. Rizzoni, Giorgio, Yue-Yun Wang, and Sergey Drakunov, "Estimation of Engine Torque Using Nonlinear Observers in the Crank Angle Domain," DSC Advanced Automotive Technologies Proceedings of the 1995 ASME International Mechanical Engineering Congress and Exposition 12-17 Nov 1995, San Francisco, CA, v 56 pp. 189-193.

16. Filipi, Zoran S., and Dennis N. Assanis, "A Nonlinear, Transient, Single-Cylinder Diesel Engine Siumulation for Predictions of Instantaneous Engine Speed and Torque," ICE Advanced Engine Simulations Proceedings of the 1997 Spring Technical Conference of the ASME Internal Combustion Engine Division 27-30 Apr 1997, Fort Collins, CO, v 28 pp. 61-70.

17. Bell, J. E., "Assessment of Diesel Engine Condition Using Time Resolved Measurements and Signal Processing," Naval Postgraduate School, Monterey, CA, M. Sc. Thesis, Sep 1996.

18. Hudson, James W. "Development and Calibration of a Torsional Engine Model for a Three-cylinder, Two-stroke Diesel Engine," Naval Postgraduate School, Monterey, CA, M. Sc. Thesis, Dec. 1997.

19. Detroit Diesel Series 53 Service Manual, Detroit Diesel Corporation, May 1990.

20. Engine Cycle Analyzer Operator's Manual, Superflow Corporation, 1990.

21. Incremental Rotary Encoders, Heidenhain Corporation, April 1997, pp. 18-19 and 32-33.

22. Kwon, Young W., and Hyochoong Bang, *The Finite Element Method Using MATLAB*, CRC Press, 1997.

23. Equivalent Mass Elastic System 3-53T Engine, Detroit Diesel Corporation, correspondence 12May1997.

24. Wilson, W. K., *Practical Solution of Torsional Vibration Problems, Volume One: Frequency Calculations,* Third Edition, Revised, John Wiley & Sons, Inc., 1956.

25. Dimensional Drawing for part #5116028 (Series 3-53 Crankshaft), Detroit Diesel Corporation, 06Mar1957, revisions through 13May1981.

26. Inertia Data for Water Brake Dynamometer, Superflow Corporation, Facsimile, 12Aug1997.

27. Carter, B. C., "An Empirical Formula for Crankshaft Stiffness," Engineering, 13 July 1928, p. 36.

28. Heywood, John B., *Internal Combustion Engine Fundamentals*, McGraw-Hill Publishing Co., 1988.

29. Taylor, Charles Fayette, *The Internal Combustion Engine in Theory and Practice*, Revised Edition, The M.I.T. Press, 1985.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Department Chairman, Code ME 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5101

4. Professor Knox T. Millsaps, Jr., Code ME/MI 4
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5101

5. Curricular Officer, Code 34 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5101

6. LT William J. Swanson 4
   539 West Third St
   Hastings, MN 55033

7. Detroit Diesel Corporation 1
   Mr. Kent Gregorius
   13400 Outer Drive, West
   Detroit, MI 48239-4001